



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

AUTONOMNÍ ŘÍZENÍ ŽALUZIÍ

AUTONOMOUS CONTROL OF SUNBLINDS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VIKTOR KOVAŘÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN VIKTORIN

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Kovařík Viktor**

Obor: Informační technologie

Téma: **Autonomní řízení žaluzií**

Autonomous Control of Sunblinds

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s vhodnými senzory umožňujícími detekovat změny počasí (např. bouřky, blesky, vítr, sluneční svit).
2. Seznamte se s vybraným aktivním prvkem pro řízení žaluzií.
3. Navrhněte vhodný způsob komunikace mezi senzory, aktivním prvkem a řídicí jednotkou (Arduino, Raspberry PI, apod.).
4. Navrhněte princip autonomního řízení žaluzií na základě informací ze senzorů. Zaměřte se na ergonomii prostředí, ve kterém je systém instalován a zohledněte zajištění ochrany žaluzií proti zničení.
5. Implementujte navržené řešení.
6. Zhodnoťte dosažené výsledky a diskutujte možná pokračování projektu.

Literatura:

- Dle pokynů vedoucí.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Viktorin Jan, Ing., UPSY FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
60200 Brno, Božetěchova 2



prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Cílem této bakalářské práce bylo seznámit se s vhodnými senzory umožňující detekovat změny počasí, dále bylo potřeba seznámit se s vybraným aktivním prvkem pro řízení žaluzií. V první části práce jsou popsány jednotlivé senzory, mikrokontroléry, které s něma mohou pracovat, router Turris, který slouží jako řídicí jednotka a komunikační protokoly. V rámci implementační části byl navrhnut princip autonomního řízení žaluzií a realizován na žaluzích od firmy Somfy. V závěru práce jsou shrnuty dosažené výsledky.

Abstract

The aim of this bachelor thesis was to acquaint with sensors which are suitable to detect changes of weather, and also with active element of blinds controller. In the first part of the thesis are described individual sensors, microcontrollers that can work with them, Turris router, which serves as a control unit and communication protocols. In the implementation part, the principle of autonomous blinds management was designed and implemented on Somfy blinds. At the end of the thesis are summarized the achieved results.

Klíčová slova

autonomní žaluzie, Somfy, automatizace, ESP8266, Arduino, Turris router, Python, meteorostanice, senzory

Keywords

autonomous sunblinds, Somfy, automation, ESP8266, Arduino, Turris router, Python, weather station, sensors

Citace

KOVAŘÍK, Viktor. *Autonomní řízení žaluzií*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Viktorin

Autonomní řízení žaluzií

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Viktorina. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Viktor Kovařík
16. května 2018

Poděkování

Rád bych poděkoval mému vedoucímu bakalářské práce panu Ing. Janu Viktorinovi za odbornou konzultaci při realizaci této bakalářské práce. Poděkoval bych také kamarádovi Tomášovi Mikulcovi za propůjčení 3D tiskárny pro jednu část práce a Přemyslu Bistrému za asistenci při kalibraci senzoru rychlosti větru s pomocí automobilu. Dále bych poděkoval rodině za odbornou pomoc při instalaci meteostanice na střeše a posléze za pomoc při testování funkčnosti. Děkuji také Mgr. Janu Knotkovi za podněty a zpětnou vazbu k dosaženým milníkům.

Obsah

1	Úvod	3
2	Požadavky	4
2.1	Cílové prostředí	4
2.2	Žaluzie	5
2.3	Systém řízení	6
3	Senzory, zařízení a komunikační protokoly	7
3.1	Sběrnice	7
3.1.1	I ² C	7
3.1.2	SPI	8
3.1.3	Analogová komunikace	8
3.2	Použité senzory	9
3.3	Zařízení	10
3.3.1	Arduino	11
3.3.2	ESP8266	12
3.3.3	Raspberry Pi	13
3.3.4	Router Turris	15
3.4	Bezdrátová rozhraní a komunikační protokoly	15
3.4.1	Bezdrátová komunikace	15
3.4.2	Komunikační protokoly	15
4	Systém žaluzií Somfy	17
4.1	Žaluzie Somfy RTS	17
4.2	Komunikační protokol Somfy RTS	17
5	Návrh řešení	21
5.1	Dílní části systému	21
5.1.1	Mikrokontrolér	21
5.1.2	Senzor	22
5.1.3	Vysílač 433 MHz	23
5.1.4	Řídicí jednotka	23
5.2	Komunikace mezi zařízeními	23
5.2.1	Logická struktura	23
5.2.2	Pravidla	24
5.2.3	Návrh tříd AutomationRule a state	26
6	Implementace	27

6.1	Příprava domu a žaluzií	27
6.2	Výběr vhodných jednotek ESP8266	28
6.3	Výroba desky pro meteostanici	28
6.4	3D tisk držáku desky v meteostanici	31
6.5	Umístění na meteostanici	31
6.6	Výroba desky ovladače žaluzií	33
6.7	Programování mikrokontrolérů	35
6.7.1	Knihovna WifiManager	35
6.7.2	Knihovna ArduinoOTA	36
6.7.3	Programování senzorové jednotky	37
6.7.4	Programování ovládání žaluzií	38
6.8	Implementace řídicí jednotky	38
6.8.1	MQTT broker	39
6.8.2	Automatizační program	39
6.9	Ověrování validity senzorů	40
6.9.1	Senzor rychlosti větru	40
6.9.2	Senzor směru větru	41
6.9.3	Senzor strážek	41
6.9.4	BMP280	42
6.9.5	TSL2561 a VEML6070	43
6.9.6	Senzor blesků AS3935	43
7	Nasazení a testování	45
7.1	Instalace systému	45
7.2	Testování funkčnosti systému	46
7.3	Výsledky z reálného používání	47
8	Závěr	49
	Literatura	51
A	Obsah přiloženého CD	52

Kapitola 1

Úvod

Současné trendy ve stavebnictví a rozvoj elektroniky nám umožňuje využívat moderní technologie k mnoha obvyklým činnostem v domácnosti.

Avšak často nejsou tyto prvky dotáhnuty do použitelného stavu. Dostupná řešení jsou často uzavřená (uživatelé jsou odkázáni na hardware a software od výrobce, případně partnerů), neúplná (na trhu chybí různé senzory, které bychom mohli využít) nebo neřeší rozšiřitelnost o další senzory.

Uvažme například dům s ovladatelnými žaluziemi, například u žaluzií od firmy Somfy lze dokoupit časovaný ovladač, kterým se může na určitou denní dobu nastavit otevření či uzavření rolety. Dále pak je možné koupit jednotku obsahující senzor rychlosti větru pro uzavření žaluzií v případě velkého větru a senzor světla, kdy při poklesu intenzity světla se zatáhnou žaluzie. Tyto senzory se chovají jako black box, kdy sice jde nastavit na těchto zařízeních, při jaké intenzitě světla či rychlosti větru se mají například markýzy schovat nebo žaluzie zatáhnout, avšak tento senzor žádné naměřené údaje neposkytuje a pouze vysílá příkazy (chová se jako ovladač reagující na rychlost větru)¹. Tyto senzory nejsou tím pádem příliš inteligentní a často nejsou nejlevnější².

Cílem této práce je proto návrh systému pro řízení žaluzií v domácnosti na základě dat ze senzorů pro různá rozhraní použitelná s zařízeními typu Arduino či Raspberry Pi. Tyto senzory jsou často velmi levné, avšak je problematické jejich využití pro obvyklé uživatele. Senzory potřebují ke své činnosti nějaký mikrokontrolér, který z nich bude číst data a reagovat na ně, popřípadě je bude posílat ke zpracování do řídicí jednotky.

Práce je rozdělena do několika částí. V úvodu jsou představeny senzory a komunikační rozhraní I²C nebo SPI či analogové, jejich možnosti a funkcionality. V další části jsou ukázána zařízení, která dané senzory obsluhují a jejich výhody a nevýhody a také případné další alternativy. Také je popsán komunikační protokol využitý pro komunikaci zařízení s centrálním serverem, kde probíhá automatizace chování. V druhé části textu práce potom je vysvětlen návrh mého řešení, jeho implementace a testování. Na závěr práce připadá shrnutí a diskuze o možnosti obecného využití řešení.

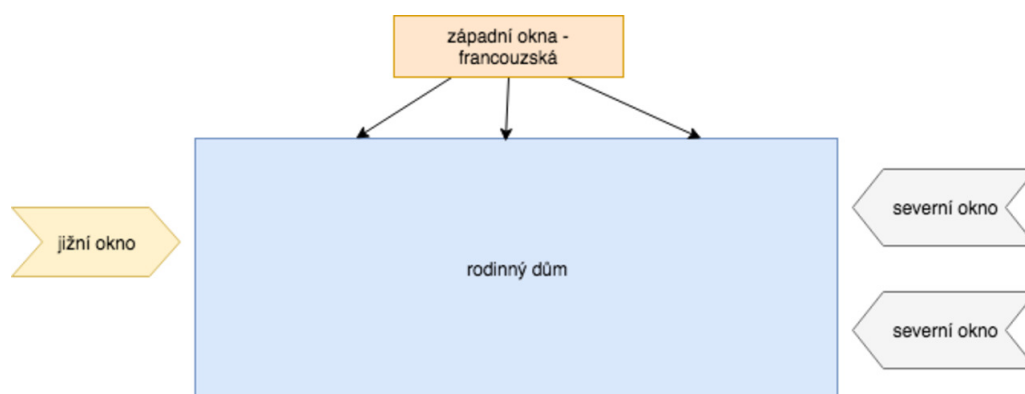
¹<https://www.somfy.cz/products/1818140/eolis-rts>

²<https://www.zaluzie24.eu/soliris-senzor-aquatic-rts-415>

Kapitola 2

Požadavky

Práce popisuje, jakým způsobem je možné v rodinném domě či firmě využívat podomítkové žaluzie s motorem dosáhnout automatizace jejich ovládání bez nutnosti investovat velké částky do nového systému nebo nákupu řešení drahého řešení, které někdy ani neposkytuje žádné údaje o naměřených hodnotách.



Obrázek 2.1: Schéma domu a orientace oken

2.1 Cílové prostředí

Rodinný dům využitý pro realizaci této práce je orientovaný velkými francouzskými okny na jihozápad, jedním oknem na jih a dvěma okny na sever, blízko kterých se často nachází květiny, které potřebují co nejdelší dobu svitu denního světla, což například v zimě je často problém, a tudíž se jeví jako nejlepší možnost tyto žaluzie otevírat hned, jakmile vyjde slunce a zase je zatáhnout večer, jakmile již slunce zapadá.

V létě je však potřeba tyto žaluzie částečně (stačí do poloviny) zatáhnout. Je to kvůli orientaci domu a velikosti oken, kdy dochází k velkému nárustu teplot v domě (v létě až přes 40 °C). Toto škodí nejen květinám, ale i obyvatelům a vybavení. U citlivých květin navíc může dojít přímým svitem slunce převážně v odpoledních hodinách k spálení listů (slunečnímu úpalu)¹.

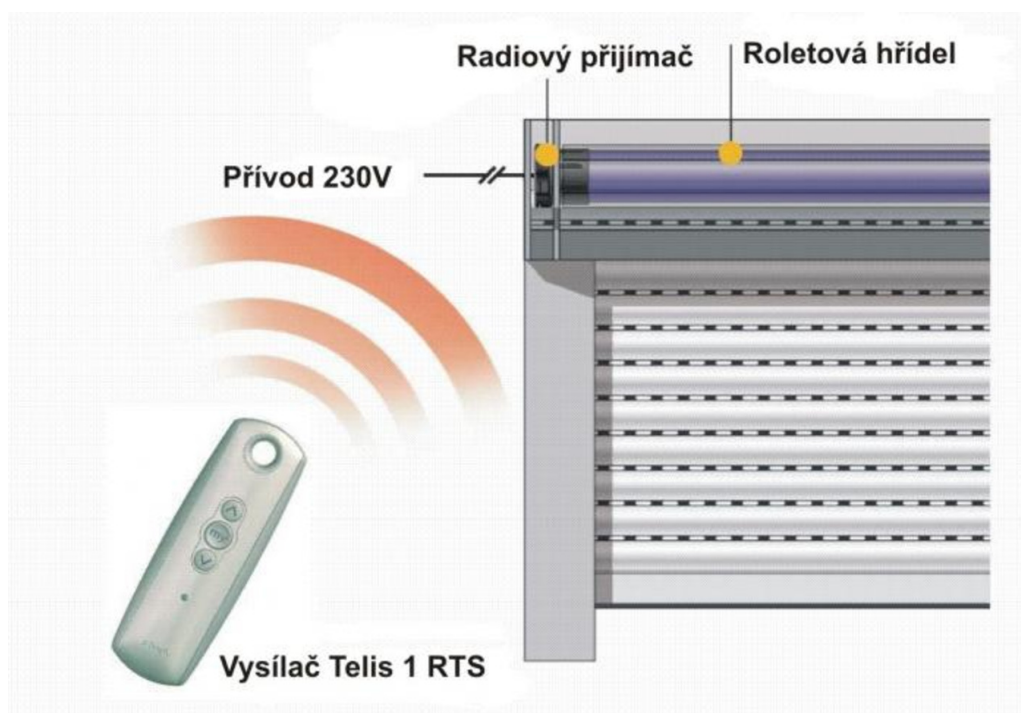
¹<https://www.ceskestavby.cz/clanky/ktere-pokojovky-na-kterou-svetovou-stranu-24142.html>

2.2 Žaluzie

V práci jsou využity žaluzie od firmy Somfy, které obsahují bezdrátovou řídicí jednotku pro jejich ovládání (obrázek 2.2). V domě je jich přítomno v přízemí celkem 6, ze kterých 3 jsou francouzská okna orientovaná na jihozápad, jedno okno na jih a dvě na sever.

Původní ovládání je realizováno dvojicí ovladačů, které obsahují každý 5 kanálů. Z nichž u každého ovladače první 3 kanály slouží pro 3 okna (každý zvlášť pro každé okno), kdy žaluzie blízko sebe mají stejný ovladač. Čtvrtý kanál je vždy nevyužitý a pátý kanál ovládá všechny 3 žaluzie naráz (u pátého kanálu svítí všechny diody signalizující zvolené žaluzie najednou, tudíž je proto logické jej takto využívat).

Nově budou žaluzie řízeny také na základě naměřených údajů ze senzorů, avšak bude zachováno i původní ovládání ovladači.



Obrázek 2.2: Bezdrátové žaluzie Somfy².

²Převzato z <https://www.universtech.cz/rolety-do-oken-okenni-rolety/predokenni-rolety/ovladani/>

2.3 Systém řízení

Systém by měl podporovat přidání libovolného senzoru a data z mikrokontrolérů mohou být zpracovávány i úplně odlišnými řídicími jednotkami autonomních systémů. Řídicí jednotka bude shromažďovat vyhodnocené údaje, ukládat je a případně zobrazovat ve webovém rozhraní. Na základě definovaných pravidel a dat ze senzorů se bude ovládat aktivní prvek žaluzií.

V tomto případě se žaluzie by měly řídit na základě denní doby v kombinaci s intenzitou světla, kdy dochází k zatažení či roztažení žaluzií.

Mimo to, pokud je venku i vevnitř vysoká teplota a senzory naměří velkou intenzitu západního slunce, žaluzie se zatahnou do poloviny, pro ochránění květin proti spálení listů a celkově celého domu před příliš vysokou teplotou uvnitř.

V případě velkého větru, bouřky či nárazového deště ze západu, jihu či severu dochází k preventivnímu zatažení žaluzií (ať už některých nebo všech podle typu či intenzity projevu počasí), ať je jakákoliv denní doba, z důvodu, že tyto povětrnostní podmínky mohou poškodit nejen žaluzie, ale také i skleněné tabule v oknech, kdy například při extrémním větru docházelo k poškození od drobných částic unášených větrem. Spuštěné žaluzie dolů také chrání okno, před jeho poškozením z důvodu velkého větru, kdy při nedovřeném nebo úplně otevřeném okně dochází z důvodu.

Kapitola 3

Senzory, zařízení a komunikační protokoly

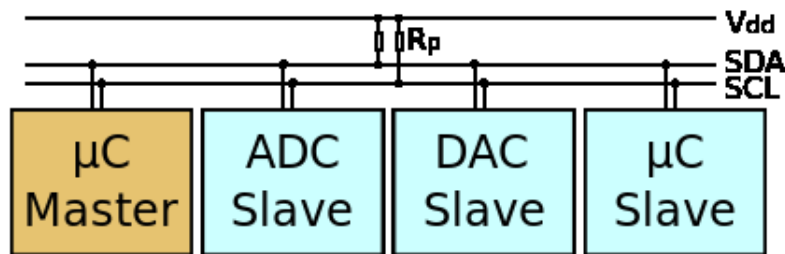
V následující kapitole jsou představeny komunikační sběrnice a senzory, které jsou využity k detekci počasí. Dále jsou zde uvedeny zařízení, které lze využít ke komunikaci s těmito senzory, případně pro automatizaci řízení žaluzií.

3.1 Sběrnice

Existuje několik rozhraní, přes která mohou senzory komunikovat. Ať už digitální využívající zejména sériové komunikace nebo analogová, kdy se měří například napětí na analogovém pinu mikrokontroléru.

3.1.1 I²C

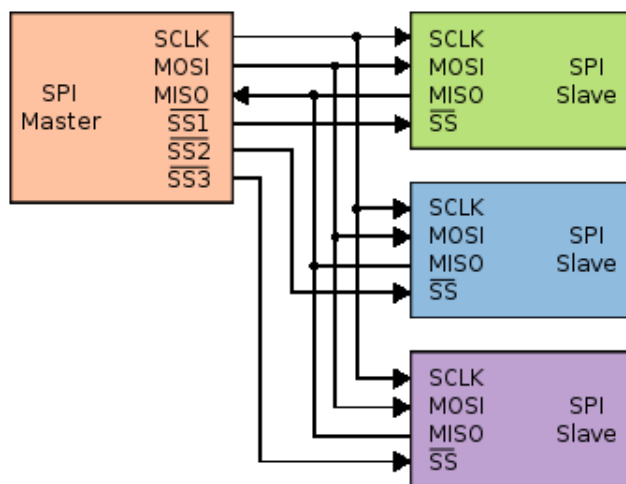
I²C (*Inter-Integrated Circuit*) je sériová sběrnice vyvinutá firmou Philips. Umožňuje připojování komponent. Komponenty pracují v režimech master nebo slave, kde master komponenty iniciují přenosy. Sběrnice dovoluje zapojit současně více master i slave komponent. Používá se k připojování komponent k vestavěným systémům i například stolním počítačům (čtení hodin RTC, čtení údajů ze senzorů v počítači či například přístup k D/A, A/D převodníkům) a telefonům. Mezi její výhodu patří například adresování, kdy pouze s pomocí dvou komunikačních vodičů (hodiny a datový vodič) umožňuje propojit až 128 různých zařízení. Komunikace je half-duplex, takže zařízení se střídají v tom, kdo posílá data [7].



Obrázek 3.1: Schéma zapojení I²C rozhraní s více zařízeními¹

3.1.2 SPI

SPI (*Serial Peripheral Interface*) je sériové rozhraní, které se podobně, jako I²C používá ke komunikaci mezi MCU a dalšími integrovanými obvody. Komunikace probíhá přes dva separátní vodiče MISO (master in slave out) a MOSI (master out slave in). Jedná se tedy v tomto případě o obousměrné spojení (full-duplex). Protože se jedná o sériovou linku, je potřeba, aby vybraná strana (v tomto případě master) generovala synchronizační signál. U sběrnice SPI se tento signál označuje jako SCLK. Adresování v tomto případě probíhá pomocí separátního pinu (nazvaný CS nebo SS), který při logické nule povolí komunikaci. Master má několik těchto výstupů, pro každé zařízení zvlášť, a to které má komunikovat, musí mít správně nastavené hodnoty pinu slave-select [7].



Obrázek 3.2: Schéma zapojení SPI rozhraní s více zařízeními²

3.1.3 Analogová komunikace

Mnoho zařízení umí díky přítomnosti AD převodníku získávat i údaje z analogových senzorů. Často se pracuje s napětím například mezi 0 až 5 V a naměřené hodnoty napětí se díky 10 bit AD převodníku (jako má například Arduino) rozdělí na 1024 hodnot.

K analogovému výstupu dat se využívá schopnosti kontroléru generovat pulzně šířkově modulovaný signál (PWM) a pokud se vhodně zvolí frekvence, tak výstupní signál může napodobit analogový.

Čtení analogových dat se využívá u zařízení, která nemají kontrolér. Příkladem může být anemometr (měří rychlost větru), na jehož výstupu je pouze napětí, které odpovídá rychlosti větru.

¹Převzato z <https://commons.wikimedia.org/wiki/File:I2C.svg>

²Převzato z https://commons.wikimedia.org/wiki/File:SPI_three_slaves.svg

3.2 Použité senzory

S ohledem na cenu, přesnost a použitá rozhraní byly vybrány následující senzory, které mohou splnit požadavky uvedené v kapitole 2.

- **Senzor teploty a tlaku BMP280**

Modul pro měření teploty a tlaku obsahuje senzor od firmy Bosch [2] a komunikuje pomocí rozhraní I²C nebo SPI.

Jeho rozsah měření je u teploty -40 až +85 °C a teplotu rozlišuje na 0,01 °C. Odchylka měření je pak +/- 1 °C a při teplotě okolo 25 °C je tato odchylka dokonce +/- 0,5 °C.

U tlaku je rozsah měření 300 až 1100 hPa, rozlišení v tomto případě je 0,18 Pa a přesnost naměřeného tlaku je +/- 100 Pa.

Senzor je napájen stejnosměrným napětím 1,8–3,6 V. Při vzorkovací frekvenci 1 Hz je jeho spotřeba 2,8 µAh a při režimu spánku jeho spotřeba klesá až na 0,1 µAh.

- **Senzor intenzity světla TSL2561**

Modul obsahuje dvě fotodiody, které zjišťují intenzitu světla ve viditelném i infračerveném spektru, a tyto hodnoty lze měřit odděleně. Díky nastavitelnému časovému rozsahu měření lze měřit intenzitu světla od 0,1 do 40 000+ Lux jednotek [6].

Senzor využívá rozhraní I²C, kde jde zvolit jednu ze tří sedmibitových adres, takže díky tomu je možnost připojit do stejného rozhraní až 3 tyto senzory.

K napájení lze využít napětí v rozmezí 2,7 až 5 V. Spotřeba v uspaném stavu je méně než 15 µA a při práci senzoru okolo 0,5 mA.

- **Senzor intenzity UVA záření VEML6070**

Tento senzor obsahuje fotodiodu, která dokáže detekovat intenzitu UVA³ záření (vlnová délka 315 nm–400 nm). Tento senzor vrací hodnoty UV záření bez patřičné jednotky, avšak v přepočtové tabulce v dokumentaci [11] lze vyčíst, které hodnoty odpovídají jaké intenzitě tohoto záření.

Obecně vzato hodnoty nad 1500 jednotek u tohoto senzoru odpovídají velmi vysokému uv záření⁴.

- **Senzor blesků AS3935**

Tento senzor obsahuje anténu a detekuje rádiové vlny způsobené bouřkou. Obsahuje proprietární algoritmus na účinné potlačování interferencí⁵ způsobených lidskými činnostmi (motory, mikrovlnné trouby).

Mezi klíčové vlastnosti patří detekce bouřky jak mezi mrakem a zemí, tak mezi mraky. Senzor umí detekovat bouřku až do vzdálenosti 40 km s rozlišením na 14 jednotek (obrázek 3.3).

³<http://theprettypimple.com/uva-versus-uvb-rays/>

⁴<https://www.meteogram.cz/uv-index/>

⁵<https://vyvoj.hw.cz/soucastky/analogove-systemy/zpracovani-signalu/jak-na-bourku-s-lc-obvodem-a-chytrym-algoritmem.html>

Table 17. Distance Estimation

REG0x07[5:0]	Distance [km]
111111	Out of range
101000	40
100101	37
100010	34
011111	31
011011	27
011000	24
010100	20
010001	17
001110	14
001100	12
001010	10
001000	8
000110	6
000101	5
000001	Storm is Overhead

Obrázek 3.3: Tabulka vzdáleností vrácených senzorem [1]

- **Anemometr z kitu meteostanice SEN-08942**

Anemometr je přístroj na měření rychlosti a případně i směru větru.

V použitém kitu se nachází mechanický anemometr, který měří rychlost větru na základě fyzického otáčení lopatek. Kromě toho obsahuje svislou lopatku pro určení směru větru.

Uvnitř části pro zjišťování směru větru je propoj, který se jednou za otočku sepne a tím propojí dva vodiče uvnitř RJ11 konektoru. Vzniklá propojení lze počítat a zjistit tím rychlost větru za určitý čas. Jedna otočka za vteřinu tohoto anemometru odpovídá rychlosti 2,4 km/h [12].

3.3 Zařízení

Pro získávání dat ze senzorů je potřeba využít nějaký kontrolér, který bude na základě senzorických dat (např. teplota, sluneční svit, ...) vyhodnocovat řízení žaluzií. K tomu se dají využít různá zařízení.

Některá jsou jednodušší (např. operačního systému). Tato zařízení se hodí na čtení dat ze senzorů a jejich předání řídicí jednotce. Pro takovéto využití se přímo nabízí mikrokontroléry jako je například Arduino či ESP8266.

V případě řídicí jednotky se často hodí, aby zařízení obsahovalo již nějaký operační systém a dokázalo pracovat například s TLS zabezpečením nebo síťovou konektivitou. Bývá tady využito také webových rozhraní či databází, které se obvykle na mikrokontrolérech problematičtěji implementují či jsou velmi zjednodušené.

3.3.1 Arduino

Arduino je rodina jednodeskových počítačů založených na mikrokontrolérech ATmega od Atmelu. Arduino Uno a Pro Mini běží na frekvenci 16 MHz a jsou doplněny 2 kB SRAM a 1 kB EEPROM a 32 kB flash paměti (verze *Mega* má všechny tyto paměti 4 krát větší). Návrh tohoto zařízení cílí na nízké pořizovací i provozní náklady a široké uplatnění. Obsahuje vstupně/výstupní rozhraní GPIO, podporuje jak digitální, tak analogové výstupy a výstupy a tím pádem je možné k němu připojit například LED diody, motory, LCD displeje, různé senzory a podobně.

Na rozdíl od například od Raspberry Pi, Arduino není úplný plnohodnotný počítač. Neobsahuje žádné grafické výstupy (pokud si je sami nenaprogramujeme) a ani podporu operačních systémů.

- **Provedení a verze**

Každá verze je mírně odlišná od ostatních, často se liší ve velikosti, způsobu napájení a počtu digitálních či analogových vstupů/výstupů.

- **Arduino Pro Mini**

Jedna z nejmenších a nejlevnějších verzí na trhu. Obsahuje mikrokontrolér založený na *ATmega328P*. Má 14 digitálních vstupně/výstupních pinů z čehož 6 se dá použít pro pulzně šířkové modulovaný (PWM) výstup. K tomu obsahuje deska ještě 6 analogových vstupů. Tento nejmenší model není vybaven USB portem ani žádným převodníkem RS-323 na USB rozhraní. Tím pádem je potřeba pro prvotní nahrání kódu připojit zařízení k například převodníku založeném na *FTDI* čipu (RS-323 to USB).

- **Arduino Uno**

Arduino Uno obsahuje stejný mikrokontrolér (*ATmega328P*), podobně jako Arduino Pro Mini. Tato verze je základní, ze které ostatní desky z této rodiny alespoň částečně vychází. Zařízení obsahuje 14 digitálních pinů pro vstup či výstup a 6 analogových. Čím se ovšem liší je ve vybavenosti konektory. Deska obsahuje jak kulatý napájecí (tzv. *barrel*) konektor, tak i USB rozhraní sloužící pro nahrávání programů. Není potřeba tedy dokupovat žádné převodníky či pájet GPIO piny a vše funguje out of box.

- **Arduino Mega**

Toto zařízení má v sobě mikrokontrolér *ATmega1280*, který má mnohem více GPIO pinů, celkem 70. 54 z nich je digitálních a z nich 15 lze použít jako *PWM* výstup. 16 pinů je vyhrazeno na analogový vstup a stejně jako Arduino Uno obsahuje USB i kulatý nabíjecí konektor.

- **Programování**

Programy se programují nejčastěji v jazyce C/C++ a externích funkcích, určených speciálně pro Arduino kompatibilní zařízení.

Obvyklý Arduino kód se skládá z funkcí `setup()`, kde se většinou programuje kód, který se spustí pouze jednou hned po spuštění kontroléru. Dále se využívá funkce `loop()`, která je periodicky volaná a slouží pro kód, který má neustále běžet. Tyto dvě části musí být v programu přítomny vždy, jinak se nepodaří program přeložit.

- **Rozšíření platformy Arduino**

Arduino podporuje celou řadu rozšiřujících modulů zvanou *shield*. Pomocí těchto modulů se dá jednoduše bez nutnosti pájení přidat různé funkcionality do tohoto *microcontroller unit* (MCU). Mezi často používaná rozšíření patří například *Ethernet Shield*, *Wifi Shield* a *GPS Shield*, které rozšiřují Arduino o možnost získávání dat z externích zdrojů. Taky se dá připojit k Arduino lithiová baterie přes battery shield. Některé shieldy využívají všechny piny Arduina, jiné pouze některé. Shieldy také často piny GPIO do kterých jsou zapojeny vedou na svůj výstup. Tím pádem je možné k Arduino stále připojit vlastní zařízení. Komunikace mezi rozšiřující deskou a MCU probíhá například přes standardy SPI nebo I²C (tady se uplatní právě to, že shieldy často mají na sobě GPIO z Arduina).

3.3.2 ESP8266

ESP8266 je jeden z velmi levných a široce dostupných Wi-Fi mikrokontrolérů. Celé ESP8266 je vlastně programovatelný Wi-Fi čip založený na RISC architektuře, kde je možnost přímo nahrát kód v LUA, MicroPythonu či C/C++ pro Arduino. Obvykle bývá podobně jako Arduino opatřen alespoň vstupně/výstupním GPIO rozhraním. Tím pádem po hardwarové stránce podporuje například LED diody, motory, LCD displeje, senzory a podobně.

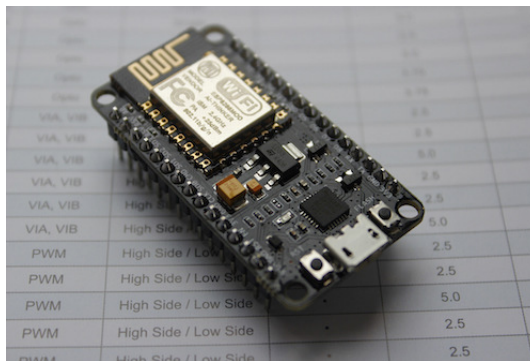
Není to plnohodnotný počítač, jak jej chápeme, a tím pádem se programuje přímo zařízení a jeho chování.

• Provedení a verze

Jelikož se jedná o levný Wi-Fi čip, je často toto zařízení využíváno jako rozšiřující Wi-Fi modul pro nějaké zařízení anebo jako součást vývojářských kitů, kde existuje velké množství různých desek.

– NodeMCU

Byl vydán v roce 2014, jako jeden z prvních vývojových kitů. Funguje víceméně jako referenční model, jelikož je postaven na open source návrhu a ostatní výrobci využívají tohoto referenčního návrhu k tvorbě vlastních verzí.



Obrázek 3.4: NodeMCU v 1.0 dev kit⁶

– Wemos

Známý výrobce desek obsahující ESP8266. Mezi populární jeho verze patří D1, D1 Mini nebo například nový LOLIN32 Pro. Zařízení tohoto výrobce (zejména

⁶Převzato z https://commons.wikimedia.org/wiki/File:NodeMCU_DEVKIT_1.0.jpg

D1 Mini a LOLIN32 Pro) se velmi inspirovaly návrhem NodeMCU a jsou s ním do velké míry kompatibilní.

- * D1 obsahuje 11 digitálních I/O pinů a všechny z nich umí jak signály přerušování, tak PWM, I2C, případně OneWire protokol, obsažen je taky 1 analogový vstup (0–3.2 V). Napájení je přes kulatý (barrel) port, do kterého lze přivést napětí mezi 9–24 V
- * D1 Mini je zmenšená verze původního D1, napájená přes microUSB a má rozměry 34.2 mm x 25.6 mm
- * LOLIN32 Pro není založen na ESP8266, ale na jeho nástupci ESP32, tím pádem toto zařízení kromě rychlejšího Wi-Fi modulu nabízí i podporu Bluetooth ve verzi 4.2 (včetně úsporné LE specifikace). Má také microSD slot a konektor pro připojení Li-Ion baterie.

• Programování

Nespornou výhodou tohoto mikrokontroléru pro vývojáře je dostupnost různých programovacích jazyků a softwaru. Programovat jej se může v jazyce *LUA*, *MicroPythonu* či v *JavaScriptu*.

Velmi podstatnou možností je taky kompatibilita se mnoha původními Arduino knihovnamí a podpora jakéhokoli kódu C/C++ ve Wiring prostředí pro Arduino MCU bez nutnosti větších úprav kódu pro použití s ESP8266. Navíc lze využít i Arduino IDE, kde se poté programuje úplně stejně jak na Arduino, jen se zvolí správný kontrolér v IDE a pro dostupnost Wi-Fi konektivity je potřeba zahrnout také knihovnu `ESP8266WiFi.h`.

3.3.3 Raspberry Pi

Je zařízení, který na rozdíl od výše zmíněných mikrokontrolérů obsahuje plnohodnotný počítač. Základem je (podle generace) 1–4 jádrový procesor doplněný o 256–1024 MB operační paměti. Všechny provedení obsahují výkonný grafický procesor VideoCore IV, který je plně zdokumentovaný a open source a podporující OpenGL ES 2.0, 1080p playback a dokonce i MPEG-4 [10].

Stejně jako Arduino a ESP8266 obsahuje GPIO rozhraní, ale také rozhraní CSI pro kameru a DSI pro displeje. Mimo tato speciální rozhraní obsahuje HDMI 1.3 (v edici Zero se jedná o miniHDMI), dále pak 10/100 Mbps Ethernet slot, USB 2.0 (v několika kusech nebo pomocí OTG přes microUSB sběrnici – Raspberry Pi Zero), výstup na 3,5 mm jack a také kompozitní výstup. K napájení slouží microUSB slot, případně GPIO a systém může běžet z microSDHC karty, na kterou je na zařízení obsažen slot (minimálně bootloader musí být obsažen na kartě, tudíž je povinná alespoň nějaká).

• Provedení a verze

Existuje několik různých modelů Raspberry Pi, které se liší velikostí či počtem portů. Některé modely se například již v nových generacích nevyskytují a každá generace přináší relativně výrazné změny, avšak zachovává kompatibilitu.

– Modely

- * Model A/A+ má pouze 1 USB port a neobsahuje ethernet. U A+ došlo ke zvětšení RAM paměti z 256 MiB na 512 MiB, snížení ceny (A – 25 USD,

A+ – 20 USD) a sloučení 3,5 mm jacku a cinch TV výstupu do jednoho jack konektoru. Základní A se již neprodává a A+ je možné ještě koupit.

- * Model B/B+ obsahuje ethernet, 2 USB porty v první generaci a 4 USB porty v následujících.
- * Compute module je určen pro vývojáře, kteří plánují si navrhnout svoji vlastní PCB. Compute module má totiž stejný fyzický slot jako má DDR2 SODIMM modul. V žádném případě teda není kompatibilní s tímto standardem, avšak kvůli rozšířenosti tohoto slotu byl vybrán právě tento. Compute module existuje jak v první generaci s původní HW konfigurací, tak v aktuální třetí generaci. Na rozdíl od standardních RPi má tato verze vestavěnou 4 GiB eMMC paměť (verze 3 lite má místo toho microSDHC slot) a neobsahuje žádné fyzické konektory.
- * Zero/Zero W je nová verze tohoto zařízení, kde je kladený důraz na malou velikost a nízkou cenu (5 USD). Hardwarově odpovídá Pi Zero první generaci Raspberry Pi, avšak CPU běží 1 GHz a má 512 MiB. Raspberry Pi Zero W obsahuje navíc Wi-Fi modul a prodává se za 10 dolarů.

– Generace

Generace Raspberry Pi se týkají pouze Modelu B a Compute modulu, přičemž první jmenovaný je zastoupen ve všech generacích a Compute module vyšel pouze v první a třetí generaci. Model A byl dostupný pouze v první generaci a nyní již nevychází.

- * Raspberry Pi (1) je původní Raspberry a obsahuje BCM2835 (ARM1176JZF-S) od Broadcomu, obsahuje slot na karty formátu SDHC/MMC.
- * Raspberry Pi 2 obsahuje 4 jádrový procesor BCM2836 (ARM Cortex-A7) běžící na taktu 900 MHz a je vybaveno 1 GiB operační pamětí.
- * Raspberry Pi 3 je nově osazeno 4 jádrovým 64-bitovým procesorem ARM Cortex-A53, který je taktován na 1,2 GHz a je vybaveno stejně jako předchozí generace 1 GiB RAM. Nově jsou v této verzi integrovány i Wi-Fi a Bluetooth moduly.

• Programování

Jelikož se jedná o plnohodnotný počítač, dá se na něm spustit například operační systém Raspbian (založený na distribuci Debian). Tím pádem jsou možnosti využití programovacích jazyků a prostředí takřka neomezené a díky popularitě RPi také velmi mnoho linuxových vývojářských nástrojů již přímo vychází pro ARM platformu.

• Možná rozšíření

Raspberry Pi je podobně jakou u Arduina či ESP8266 možno rozšířit různými rozšiřujícími shieldy, které přidají například podporu pro baterie, lepší audio čip nebo například různá relé a podobné věci. Tyto možnosti vyplývají pouze z vlastností univerzálnosti GPIO rozhraní a není problém tím pádem bez větších potíží zprovoznit tyto shield například na Arduinu a naopak.

Deska ale obsahuje taky rozhraní CSI a DSI, které mají vlastní konektor na RPi. CSI kamera v aktuální verzi (v2) obsahuje Sony IMX219 senzor a má rozlišení 8 megapixelů (v1 má 5 megapixelů). DSI není příliš rozšířené, navíc chybí dokumentace ke způsobu zobrazování. Tím pádem neexistuje kromě oficiálního displeje moc dalších,

cena není úplně přívětivá a komunita používá raději displeje přes HDMI port případně GPIO rozhraní.

3.3.4 Router Turrís

Jako možnou řídicí jednotku lze využít například Turrís verze 1.0 od sdružení CZ.NIC. Tento router byl součástí programu Projekt Turrís, který měl za úkol anonymně monitorovat datový provoz v domácnostech a na základě dat od uživatelů detekovat případné bezpečnostní rizika na českém internetu [3].

Kromě pravidelných bezpečnostních aktualizací nabízí tento router velmi dobré hardwarové vybavení (například 2 GB RAM, procesor PowerPC architektury), dále také firmware od CZ.NIC, jehož součástí je i OpenWRT operační systém pro routery.

3.4 Bezdrátová rozhraní a komunikační protokoly

Zde je popsáno několik možných způsobů komunikace mezi řídicí jednotkou a senzory. Využívá se bezdrátových rozhraní a komunikačních protokolů.

3.4.1 Bezdrátová komunikace

Zařízení si předávají svá data přes bezdrátová rozhraní, která umožňují je umístit i na těžko dostupná místa.

- **Radiový přenos dat**

Jak bylo zmíněno v kapitole 4, tak žaluzie využívají pro komunikaci 433 MHz frekvenci. K předávání dat tedy bude sloužit PWM signál a využije se analogového rozhraní (sekce 3.1.3) některého z kontrolérů.

- **Wi-Fi, Bluetooth**

Ke komunikaci mezi senzorem a řídicí jednotkou je možné využít například technologií Bluetooth a Wi-Fi. Oba dva standardy obsahuje například Raspberry Pi 3 už v základu, Wi-Fi pak je možné využít jak u Arduina (s Wi-Fi shieldem), tak s ESP8266.

3.4.2 Komunikační protokoly

K přenosu dat z mikrokontrolérů do řídicí jednotky je potřeba využít nějakého komunikačního protokolu.

- **HTTP + JSON**

Pomocí HTTP POST požadavku je možné jednoduše posílat data do řídicí jednotky. K posílání dat je vhodné serializovat data pomocí JSONu a později je v řídicí jednotce interpretovat. Data nejsou při HTTP přenosu zabezpečena, jsou však nenáročná na výkon koncových zařízení.

- **MQTT protokol**

Pro komunikaci mezi centrální jednotkou a senzory je možné využít i protokolu MQTT. Jak z hlediska své rozšířenosti, tak z důvodu snadné implementace a nenáročnosti pro slabší hardware.

Message Queuing Telemetry Transport, zkráceně MQTT je protokol pro předávání dat mezi zařízeními pomocí centrálního bodu, který se nazývá *broker*. Zprávy, které jsou posílány jsou roztríděné do různých témat (*topic*) a zařízení je buď to tohoto tématu odesílá (*publish*) nebo je přihlášeno k odběru daného tématu (*subscribe*) a centrální broker potom zprávy, které jsou odbírány posílá do zařízení. Každé zařízení může být subscriber i publisher zároveň i na více různých místech.

Přenášet se mohou jakákoliv binární data, často se přenášejí také JSON data či text. Maximální velikost zprávy je v aktuální verzi 256 MB, avšak pro IoT údaje jsou více než dostačující.

Témata v MQTT mají trošku podobu adresářové struktury, kdy se může nadefinovat například *topic* s názvem *domov/střecha/senzor_rychlosti_větru*. MQTT témata podporují názvy v UTF-8, tudíž není problém s diakritikou.

MQTT umí 3 vrstvy QoS, které slouží k ověření, zda data byla bezpečně doručena. Zařízení, které odbírá dané téma tudíž může poslat zpět brokerovi potvrzení, že data obdrželo. Nejvyšší vrstva QoS 2 umožňuje po potvrzení doručení ještě zprávu smazat a tím pádem již není posílána vícekrát.

Kapitola 4

System žaluzií Somfy

System žaluzií Somfy se skládá z motoru, samotné žaluzie a řídicí elektroniky, která slouží k ochraně motoru tím, že si pamatuje maximální a minimální stupeň vysunutí žaluzií, aby nedošlo k jeho zadření/protáčení. Paměť také obsahuje například uživatelsky definované oblíbené pozice. Každá žaluzie má svoji řídicí jednotu a může fungovat samostatně. Součástí řídicí jednotky je bezdrátový modul, který slouží k přijímání signálů z ovladačů.

System umožňuje připojit několik různých ovladačů ke každé žaluzii a je také možné jedním ovladačem kontrolovat více žaluzií. Pro přidání nového ovladače je potřeba ovladač se stávajícími žaluziemi spárovat (sekce 4.2). K ovládání směru žaluzií slouží tlačítka ovladače *nahoru* a *dolů*. Žaluzie se zastavují stiskem tlačítka *My* a případně se stejným tlačítkem (pokud jsou zastavené) dají umístit do uložené pozice (sekce 4.2).

4.1 Žaluzie Somfy RTS

Bezdrátová komunikační jednotka žaluzií od firmy Somfy využívá zabezpečený, jednosměrný komunikační protokol, který má podobnost v technice bezdrátového odemykání automobilů.

4.2 Komunikační protokol Somfy RTS

Zařízení pracuje na frekvenci 433,42 MHz a každý zařízený má svůj unikátní identifikátor. Dále je podstatný již zmíněný rolling code, který se musí s každým stiskem inkrementovat.

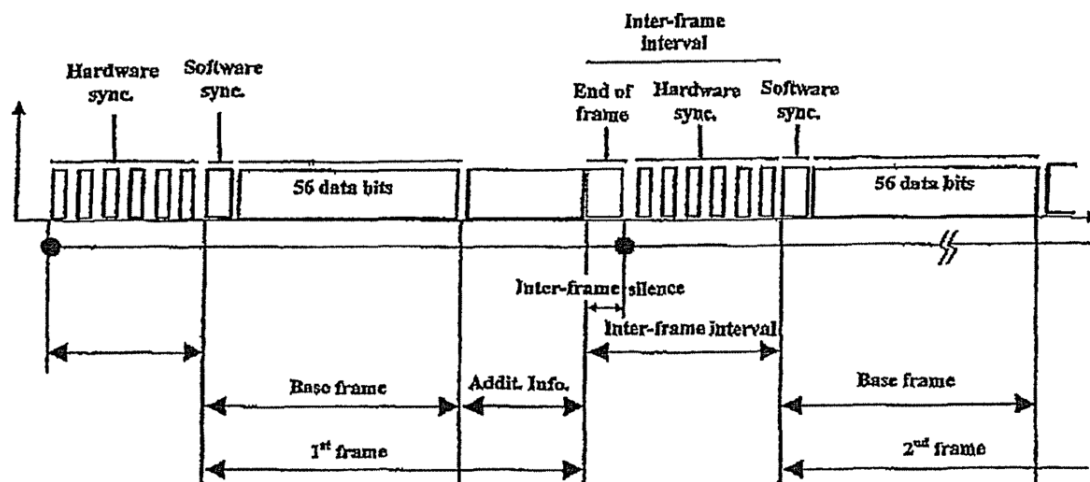
Je využito ASK/OOK¹ modulace a kódování Manchester² [9], kde náběžná hrana je logická 1 a sestupná hrana je logická 0. Délka datového rámce je 56 bitů a jeden bit je přenášen v periodě 1208 μ s.

Struktura rámce se skládá z hardwarové synchronizace, softwarové synchronizace, dat a mezerou mezi jednotlivými rámci, kdy se nevysílají žádná data. HW synchronizace je sekvence náběžná hrana, sestupná hrana, náběžná hrana, sestupná hrana, které jsou stejně dlouhé (každá hrana 2 periody, 2416 μ s). Softwarová synchronizace má nástupnou hranu dlouhou 4550 μ s a sestupnou dlouhou 604 μ s. Jeden bit dat je zakódovaný v jedné periodě 1208 μ s, celkem je na data vyhrazeno 56 bitů. Mezera mezi rámci je dlouhá 30415 μ s. Před odesláním prvního datového rámce posílá protokol 9415 μ s dlouhou probouzeckou zprávu

¹<https://www.solidremote.com/blog/what-is-ask-and-ook-modulation/>

²<https://www.allaboutcircuits.com/technical-articles/manchester-encoding-what-is-it-and-why-use-it/>

(náběžná hrana), následovanou 89565 μs pauzou (sestupná hrana). Tento probouzecí signál však není povinný a žaluzie bez něj fungují.



Obrázek 4.1: Struktura rámce u protokolu Somfy RTS, jak jej popisuje patent [5].

Časování u tohoto protokolu není kritické a údajně žaluzie tolerují 15–20 % odchylku. Taký stačí k aktivování žaluzií poslat rámec pouze jednou, přičemž v reálném ovladači se tento stejný rámec posílá neustále dokola, dokud se drží stisknuté tlačítko ovladače a duplicitní rámce, které jsou doručeny po přijetí příkazu už nemají na žaluzie vliv, protože došlo k inkrementování rolling kódu. Tím jde jen o pokus eliminování ztráty zprávy.

• Struktura dat

Data jsou uložena v 56 bitech, v prvních 8 bitech je uložen identifikační klíč protokolu, jehož hodnota musí začínat 0xA na prvních 4 bitech a na následujících 4 bitech může být libovolná hodnota (.

Následující 4 bity obsahují ovládací kódy žaluzií (kódy uvedeny v sekci 4.2). Tyto kódy jsou pevně dané a nemění se u žádného z modelů.

Další 4 bity obsahují kontrolní součet (checksum), který prvně vynulován a následně proběhne po sestavení celého datového bloku XOR po půl slabik současně s následující a poté se poslední 4 bity výsledku uloží do původního rámce na tuto pozici.

Následuje tzv. rolling code, který je uložený jako 16 bitové bezznaménkové číslo uložené jako big-endian a inkrementuje se s každým stiskem ovladače.

Posledních 24 bitů je vyhrazeno na unikátní identifikační adresu zařízení, které ovládá žaluzie. Tato adresa je uložena ve formě little-endian a slouží k rozlišení, který ovladač vysílá data. Je pak možné používat například na každou žaluzii jiný ovladač a ovládat každou zvlášť, případně ovládat jedním všechny najednou. Oficiální ovladače často obsahují funkci přepínání kanálů, kdy každý kanál se chová jako samostatný ovladač, a tím je možno ovládat každou žaluzii zvlášť nebo všechny najednou podle zvoleného kanálu.

key	ctrl cks	Rolling Code	Address(A0 A1 A3)
-----	------------	--------------	-------------------

Obrázek 4.2: Struktura dat u protokolu Somfy RTS [9].

• Ovládací kódy

Součástí datového rámce jsou ovládací kódy, které jsou odesílány žaluziím a ty následně provedou akci na základě přijatých dat.

Kód	Tlačítko	Popis
0x1	My	Zastaví žaluzii nebo ji posune do „oblíbené“ pozice (4.2)
0x2	Nahoru	Pohyb žaluzie nahoru
0x3	My + Nahoru	Nastavení maximální horní pozice motoru (zvednutí)
0x4	Dolů	Pohyb žaluzie dolů
0x5	My + Dolů	Nastavení maximální spodní pozice motoru
0x6	Nahoru + Dolů	Přejde do nastavení limitů motorů (prvotní nastavení)
0x7	—	—
0x8	Prog	Přepíná žaluzie do režimu (od)registrování ovladačů (4.2)
0x9	Sun + Flag	Povoluje sluneční detektor (jen ovladač Telis Soliris RC)
0xA	Flag	Zakazuje sluneční detektor

• Párování

Ovladače lze k žaluziím přidávat a odebírat v libovolném množství. Jeden ovladač může být spárován s několika žaluziemi a jedna žaluzie může mít přiřazeno několik různých ovladačů.

– Registrace ovladačů

Přidávání nových ovladačů do systému se dělá buď pomocí resetu zařízení do výchozího nastavení nebo za pomoci již spárovaného ovladače, a to stiskem tlačítka *PROG* na původním ovladači a následně stejným postupem na novém ovladači. K jednomu ovladači je možné přiřadit libovolný počet žaluzií a také většina modelů jak nástěnných tak přenosných ovladačů obsahuje několik kanálů (každý kanál v odesílaném datovém rámci je reprezentován jako unikátní ovladač, více v sekci 4.2) a tím pádem je možné ovládat jedním ovladačem i jen nějakou skupinu žaluzií.



Obrázek 4.3: Ovladač žaluzií Somfy TELIS 4 RTS³.

– **De-registrace ovladačů**

Odregistrace se dělá úplně stejně za pomoci dvou ovladačů, kdy se na ovladači, který má zůstat spárovaný, stiskne *PROG* tlačítko a následně na ovladači určenému k odebrání stiskne také tlačítko *PROG*.

Taky je možné úplně resetovat celé žaluzie pomocí odpojení přívodu elektřiny k motoru na 3 vteřiny, následně obnovení přívodu po dobu osmi vteřin, a nakonec po dobu 3 vteřin odpojení zdroje elektřiny znovu. Poté bude paměť žaluzií kompletně smazána a motor bude čekat na *PROG* tlačítko nového ovladače.

• **Oblíbená pozice žaluzií**

Žaluzie podporují nastavení pozice rolet do uživatelem definované pozice „oblíbené“, kdy se může do vnitřní paměti žaluzií uložit pozici, kam přesně mají žaluzie jet při vyvolání této funkce. Například pokud bychom chtěli jedním tlačítkem přikázat žaluzii se zatáhnout v určitém stádiu otevření bez nutnosti manuálního zastavování. Toto nastavení se dělá tak, že po nastavení manuální pozice žaluzií se podrží tlačítko *My* dokud se žaluzie nepohnou nahoru a dolů. Potom pokud chceme, aby žaluzie dojezly do uložené pozice, stačí při zastavených žaluziích stisknout tlačítko *My* znovu.

³Převzato z <http://charmblinds.com/offices/>

Kapitola 5

Návrh řešení

V následující kapitole je popsán návrh, jakým způsobem budou zařízení mezi sebou propojena a jakým způsobem bude probíhat automatizace.

K propojování senzorů a žaluzií je možné využít například mikrokontrolér ESP8266, který přes Wi-Fi komunikuje s centrální řídicí jednotkou. Místo ESP8266 je možné taky využít například Arduino opatřené bezdrátovým modulem případně i Raspberry Pi, které, je ale zbytečně komplexní pro pouhé čtení dat ze senzorů a jejich předávání.

Řídicí jednotka může být například počítač (ať už domácí desktopový server nebo například již zmíněné Raspberry Pi), kde probíhá vyhodnocení dat ze senzorů a je rozhodováno o ovládání žaluzií.

Mimo se zvažuje využití několika senzorů jako například senzor intenzity světla, senzor blesků, senzor teploty a tlaku nebo například anemometr. Tato zařízení jsou více popsána v sekci 3.2.

5.1 Dílčí části systému

Systém jako celek se sestává z různých částí, které jsou mezi sebou propojeny. V této sekci je tedy uvedeno, co k čemu je připojeno.

5.1.1 Mikrokontrolér

Pro obsluhu senzorů byl zvolen mikrokontrolér Wemos D1 Mini založený na ESP8266 a Wemos ESP-WROOM-02 (jedná se o identický mikrokontrolér, ale je vybaven slotem pro lithiovou baterii typu 18650 a nabíjecím obvodem). První z nich je v této práci určen pro ovládání žaluzií, z důvodu lepšího dosahu a druhý uvedený se nachází na střeše uvnitř meteorologické stanice.

Ve venkovní jednotce není vhodné mít jednotlivé senzory neupevněné a nezaizolované. Navíc senzory meteostanice SEN-08942 se připojují pomocí R14 konektorů, tudíž je vhodné pro ně vyrobit slot. Navíc je nejspíše potřeba zajistit určitou modularitu (pokud by došlo k poruše nějakého senzoru, aby jej šlo vyměnit za jiný kus bez nutnosti rozebírat celou meteostanici).

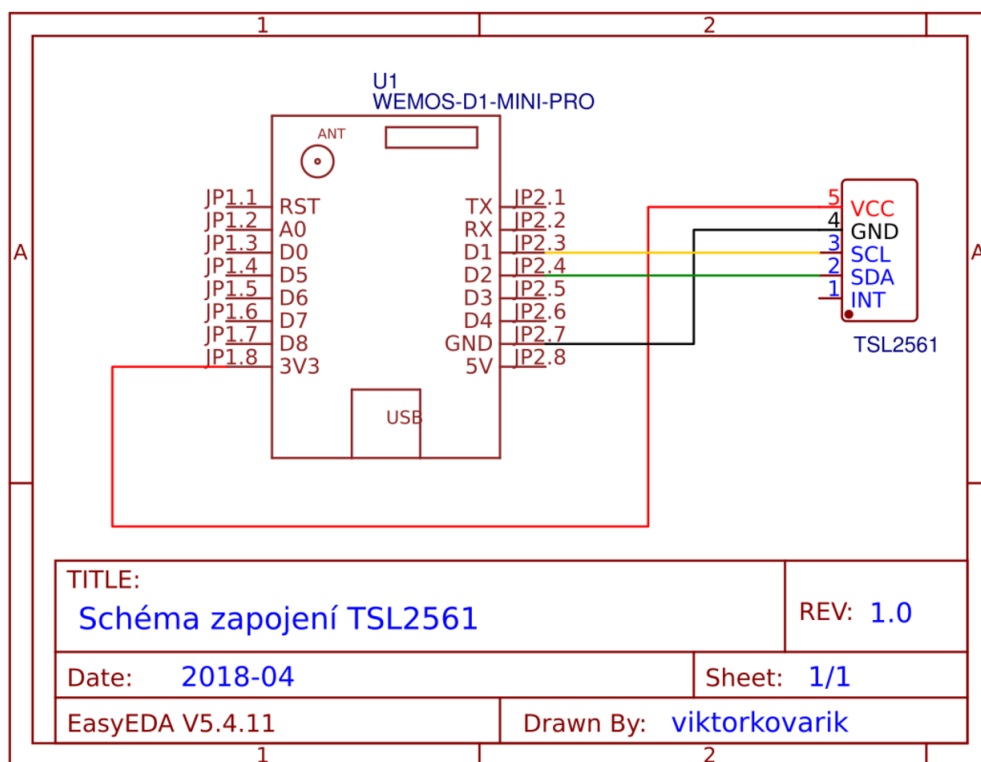
Z toho důvodu bylo rozhodnuto vyrobit pro oba mikrokontroléry desky plošných spojů, které budou mít sloty pro všechny potřebné senzory a v případě poruchy bude možno vyměnit pouze daný vadný senzor nebo mikrokontrolér a vyměnit jej za jiný kus. Pro venkovní jednotku je navíc součástí meteostanice SEN-08942 je i krytka, která vhodně zakryje senzory včetně mikrokontroléru, stačí tedy pouze vyrobit spodní držák a umístit

desku do něj. U vnitřní jednotky ovladače žaluzií je možné také malou desku a umístit ji do nějaké krabičky, což umožní jednodušší manipulaci.

5.1.2 Senzor

Celkově je zvoleno 8 různých senzorů, 5 z nich jsou pro rozhraní I²C a tím pádem je možné je připojit najednou k jednomu mikrokontroléru.

Zbylé 3 jsou analogové. Senzor srážkového úhrnu a rychlosti větru je založen na sepnutí svých dvou pinů, kdy po jejich sepnutí lze například sčítačkou nebo signálem přerušení zjistit počet těchto událostí a na základě údajů z dokumentace a vlastní kalibrace lze pak určit daná událost. Poslední analogový senzor je směrová růžice pro určení směru větru. Senzor obsahuje 16 rezistorů uložených kolem osy otáčení, které při použití děličky napětí s vhodným rezistorem může na vstupním pinu akceptovat referenční napětí (například 3.3 V u ESP8266) a na výstupním pinu lze poté změřit změnu způsobenou rezistorem uvnitř. Toto měření pak lze převést na 16 světových stran, případně na úhel natočení růžice ve stupních (ve skocích po 22,5°).



Obrázek 5.1: Schéma zapojení senzoru TSL2561 k zařízení Wemos D1 Mini

5.1.3 Vysílač 433 MHz

Žaluzie ke svému ovládání potřebují 433 MHz vysílač. Pro tento účel byl vybrán vysílač XY-FST, pro svoji jednoduchost, nízkou cenu a široké provozní napětí (může pracovat na 3–12 V stejnosměrného napětí). Obsahuje 3 piny, dva pro napájení (VCC, GND) a jeden datový. Na tento pin je možné připojit PWM výstup z mikrokontroléru a tím pádem přenášet příkazy žaluziím.

5.1.4 Řídicí jednotka

Řídicí jednotkou byl zvolen router Turris (sekce 3.3.4). Router obvykle běží nepřetržitě a pro svou podporu LXC kontenerů [4], v nichž je možné provozovat několik instancí například linuxové distribuce Debian, byl vybrán jako vhodná řídicí jednotka pro toto zadání.

5.2 Komunikace mezi zařízeními

Jelikož senzorů je několik a nachází se na různých místech, je potřeba data nějakým způsobem předávat. K získávání dat se senzorů lze použít zařízení zmíněná v kapitole 3.3 s využitím bezdrátového přenosu (sekce 3.4.1) a komunikačních protokolů (3.4.2).

Pro svoji jednoduchost, nenáročnost, univerzálnost a popularitu byl vybrán pro komunikaci protokol MQTT 3.4.2.

Broker a řídicí software poběží na řídicí jednotce. K brokeru se budou připojovat jednotlivé mikrokontroléry i řídicí software [8].

Jednotlivé senzory budou mít svůj vlastní téma například `sensor/temperature` do nějž je publikován JSON výstup z MCU obsahující naměřenou hodnotu, typ senzoru i jednotku.

Řídicí software odebírá témata, do kterých je publikováno, vyhodnocuje doručená data a na jejich základě pošle zprávu například do tématu `device/sunblind_all`, který odebírá mikrokontrolér s připojeným vysílačem 5.1.3.

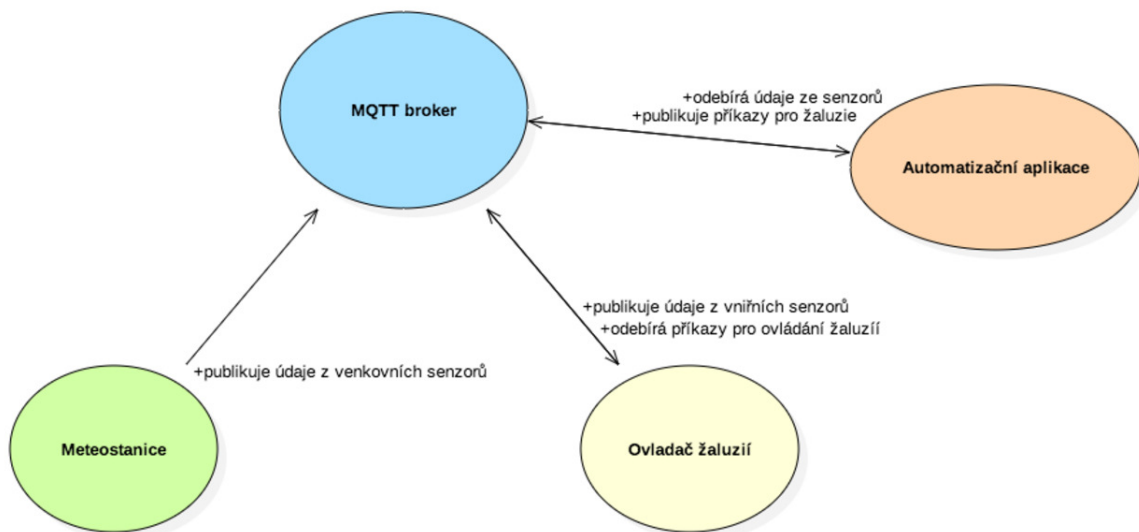
5.2.1 Logická struktura

Logická struktura řídicí jednotka se skládá z MQTT brokeru a samotného automatizačního programu.

Senzory na meteostanici a vnitřní senzory umístěné na ovladači žaluzií odesílají (publikují) svá data do různých témat podle typu senzoru.

Témata, do kterých je publikováno jsou následující:

- `sensor/temperature` — venkovní teplota
- `sensor/temperature_inside` — teplota uvnitř domu
- `sensor/pressure` — tlak vzduchu
- `sensor/pressure_inside` — tlak vzduchu uvnitř domu
- `sensor/luminosity` — počet luxů světla venku
- `sensor/uv` — intenzita venkovního uv záření
- `sensor/lightning` — v případě bouřky je zde publikována zpráva o její vzdálenosti
- `sensor/windspeed` — rychlost větru



Obrázek 5.2: Schéma propojení MQTT brokeru s ostatními částmi

- sensor/winddirection — směr větru ve stupních
- sensor/rainfall — počet milimetrů srážek za hodinu

5.2.2 Pravidla

Pro autonomní řízení žaluzií byla navržena pravidla, která zajišťují alespoň částečnou ochranu okna před zničením z důvodu špatných povětrnostních vlivů, ochranu květin před horkem či silným slunečním svitem a automatické ovládání žaluzií na základě denní doby.

Následující pravidla jsou sesazena od nejvyšší priority po nejnižší. Jak již bylo zmíněno v sekci 6.8.2, pravidla s nejnižší prioritou platí téměř vždy a v případě, že přijde nějaká událost s vyšší prioritou, převezme ovládání příslušné pravidlo a po jeho zneplatnění se žaluzie nastaví dle nižších priorit.

NearStormPreventionRule

Pravidlo vyhodnotí, zda je bouřka v blíže než 5 kilometrů od domu a pokud ano, zavře všechna okna po dobu uloženou v konstantě `LIGHTNING_COOLDOWN` (výchozí hodnota 10 minut). Po vypršení doby pravidlo přestává platit.

FarStormPreventionRule

Pravidlo pro vzdálenější bouřky, u blesků blíže než 15 kilometrů preventivně zavře žaluzie do poloviny oken a v případě, že nějakou dobu nedošlo k dalším bouřkovým událostem, toto pravidlo se zneplatní a je předáno řízení pravidlu s nižší prioritou.

VeryFastWindPrevention

V případě prudkého (velmi silného) větru (cca 50 km/h¹) dojde k uzavření všech oken, dokud tato rychlost větru se nesníží. Nepočítá se rychlost okamžitá, ale průměrná rychlost větru vycházející z počtu naměřených hodnot podle konstanty `NUMBER_OF_MEASUREMENTS` (v případě výchozí hodnoty se jedná o průměrnou rychlost větru za posledních zhruba 5 minut).

WindPrevention

Pravidlo pro silný vítr (39 km/h a více) do půlky zavře pouze žaluzie, na které vítr vane. V tomto případě se tedy využívá převážného směru větru za poslední dobu a toto pravidlo slouží zejména k tomu, aby otevřená okna zasažená větrem nebyla poškozena.

StrongRainPrevention

Pokud dojde k velmi silné intenzitě srážek (srážkový úhrn vyšší, než je 40 milimetrů za hodinu²), dochází k úplnému uzavření oken všech do doby, než přestane pršet nebo se déšť zmírní. Toto pravidlo slouží zejména k tomu, že za velkého deště může dojít k namočení podlah v domě nebo případných elektrických zařízení u oken, pokud není okno řádně uzavřeno.

RainPrevention

Toto pravidlo se aktivuje v případě silného deště (cca 8 milimetrů za hodinu), aby došlo k ochraně domu před vnikem vody otevřenými okny. Dochází k uzavření všech oken do půlky. Rovněž toto pravidlo následuje předchozí pravidlo, tím, že pokud velmi silná intenzita přechází na silnou intenzitu, dojde k otevření všech žaluzií alespoň do poloviny.

HeatPreventionRule

V letních horkých dnech je potřeba ochránit žaluziemi květiny v domě před poškozením od slunce a také udržovat dům v rozumných teplotách. K tomu slouží toto pravidlo, které měří průměrnou teplotu v domě i venku, dále potom intenzitu světla a uv záření. Pokud je opravdu horký den, s vysokým stupněm uv záření a v domě teplota také překračuje povolené meze, dojde k uzavření žaluzií do poloviny. Toto má za následek ochranu před přímým slunečním světlem většiny rostlin a také před nadměrným růstem teplot.

NightCloseRule

Pravidlo zajišťující uzavření žaluzií večer, když poklesne sluneční svit pod patričnou úroveň, toto pravidlo počítá s průměrnou intenzitou světla za posledních několik minut a je aktivováno pouze ve večerních hodinách. Tím pádem se přechází například uzavření žaluzí při zastínění senzoru.

DayOpenRule

Zde dochází k opačné události než u předchozího pravidla, ráno se žaluzie pomocí tohoto pravidla samy otevrou.

¹<https://www.tpocasi.cz/meteorologicke-pojmy/beaufortova-stupnice-sily-vetru/>

²<https://cs.wikipedia.org/wiki/Sr%C3%A1%C5%BEky>

NoLumiSensorRule

V případě, že nedošly žádné údaje ze senzoru intenzity světla (například kvůli poruše), dojde ke zjištění času východu/západu slunce z internetu a poté k provedení patřičné akce.

Default

Výchozí pravidlo pokrývá dobu, kdy například jsou již večerní hodiny, avšak slunce ještě nezapadlo. Pravidlo si pamatuje poslední událost z předchozích tří pravidel a pokud dojde například v bouřce mezitím, je tímto pravidlem vrácen předchozí stav žaluzií před bouřkou (například při krátké bouřce v 17 hodin by se jinak již žaluzie neotevřely, protože z hlediska času již není den a blíží se doba uzavírání).

5.2.3 Návrh tříd `AutomationRule` a `state`

V rámci kódu je vhodné využití objektově orientovaného psaní kódu, uvažována je třída `AutomationRule`, obsahující několik metod. V první řadě metodu pro vyhodnocení, zda je pravidlo zrovna aktivní. Dále pak metodu, která určuje prioritu pravidla a v neposlední řadě metodu, která určuje, které žaluzie (nebo skupina žaluzií) mají být ovládány.

```
class AutomationRule{
    method checkConditions(state);
    method getPriority();
    method getTargetPosition();
    method getTargetBlind();
}
```

Metoda `checkConditions(state)` bude kontrolovat, zda podmínky dané pro toto pravidlo platí a vrací `true`, pokud jsou podmínky splněny. Parametr `state` je instancí třídy, která v sobě ukládá aktuální (případně i minulý) stav všech senzorů. Pokud první metoda je vyhodnocena jako pravdivá, jsou z ostatních metod získána data. Metoda `getPriority()` vrací prioritu pravidla, metoda `getTargetPosition()` říká, do jaké cílové pozice se mají žaluzie nastavit (nahoru, dolů, zavřít napůl) a metoda `getTargetBlind()` slouží k určení, které žaluzie mají vykonat toto pravidlo.

Dále je také potřeba navrhnout třídu, která bude udržovat aktuální data ze senzorů a případné další informace jako například čas poslední obdržené zprávy.

```
class State{
    double temperature;
    double windspeed;
    method add_data(temperature, windspeed);
}
```

Ve třídě `State` budou například proměnné, ve kterých bude uložena aktuální teplota a rychlost větru. Součástí této třídy bude taky metoda `add_data(...)`, která umožní proměnné plnit daty.

Kapitola 6

Implementace

V této kapitole je popsán postup implementace jednotlivých částí projektu, jako je výroba desek plošných spojů pro meteostanici i ovladač žaluzií, využití knihovny a jejich popis a další detaily vývoje.

6.1 Příprava domu a žaluzií

Rodinný dům je orientován třemi velkými francouzskými okny na západ, jedním oknem na jih a dvěma okny na sever (sekce 2.1).

Všechna tato okna v přízemí obsahují podomítkové žaluzie s motorem od firmy Somfy a jejich ovládání je realizováno dvojicí vícekanálových ovladačů (lze jedním ovladačem ovládat více žaluzií).

Každý ovladač má na prvních 3 kanálech přiřazeny žaluzie (postupně za sebou, jak jdou), čtvrtý kanál je neobsazen a pátý kanál (svítí všechny 4 indikační LED diody, čímž naznačuje, že jsou vybrány všechny žaluzie) je spárován se žaluziemi z kanálů 1–3.



(a) Dům je orientován velkými francouzskými okny na západ



(b) Detail na žaluzii

Obrázek 6.1: Dům a žaluzie

Změny provedené v zařízení domu nejsou nijak zvlášť viditelné, do žaluzií nebylo potřeba fyzicky nijak zasahovat, byly pouze přidány 7 virtuálních ovladačů, kde ke každé žaluzii přísluší ovladač s unikátní adresou 0x121311 až 0x121316, přičemž sedmý virtuální ovladač s adresou 0x121310 je spárován (sekce 4.2) se všemi žaluziemi. Také byla u všech žaluzií

definována tzv. oblíbená poloha, do které se při stisku tlačítka **My** (sekce 4.2) přesunou. Tato poloha zhruba odpovídá středně otevřené žaluzii, protože v případě horkého počasí bývají žaluzie obdobně umístěny ručně.

Zařízení ovládající žaluzie 6.6 bylo umístěno do skříně v přízemí a bylo využito volné elektrické zásuvky uvnitř k napájení. I když zařízení není umístěno uprostřed domu, tak díky přidané anténě nemá toto umístění vliv na dosah signálu a ovládání všech žaluzií.

Jednotka meteostanice 6.3 je umístěna na střeše domu na tyči poblíž paraboly satelitu a antén. Napájecí kabel je doveden na půdu domu, kde je přiveden do napájecí soustavy domu.

Obyvatelé domu byli seznámeni s provedenými změnami a faktem, že stávající ovladače fungují i nadále na manuální ovládání žaluzií a v případě, že budou chtít žaluzie nastavit podle sebe, nebude toto jejich nastavení nijak vadit v pozdějším automatizovaném chování.

6.2 Výběr vhodných jednotek ESP8266

Kvůli omezenému dosahu 433 MHz vysílače a také potřebě monitorovat kromě venkovní teploty i tu vnitřní, byly jednotky použity dvě, Wemos ESP-WROOM-02 (má stejně nazvané piny včetně jejich možností, jako má Wemos D1 Mini a mimo to obsahuje slot pro lithiovou baterii typu 18650) a Wemos D1 Mini. První z nich, zejména kvůli přítomnosti slotu na baterii byla určena do části meteostanice a Wemos D1 Mini byl umístěn do vnitřní jednotky pro ovládání žaluzií.

6.3 Výroba desky pro meteostanici

Z důvodu přítomnosti krytky na měřicí senzory v balení meteorologického KITu SEN-08942 byla vyrobena deska plošných spojů pro jednotlivé venkovní senzory a mikrokontrolér.

Při testování jednotlivých senzorů a následné odhalení jednoho vadného mikrokontroléru a senzoru BMP280 bylo rozhodnuto mít na desce místo napřímo připojených senzorů GPIO patice pro případnou jednoduchou výměnu nefunkčních částí bez potřeby pájení.

Jednotlivé komponenty jsou mezi sebou propojeny podle následujícího schématu.

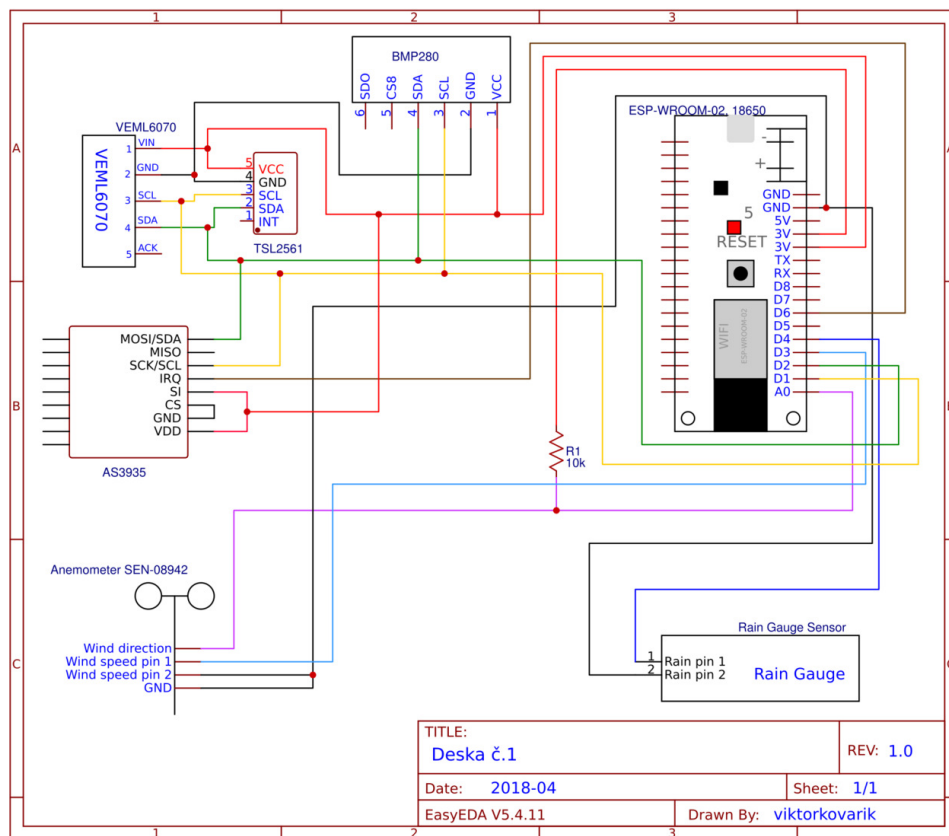
Deska byla vyrobena ze dvou jednostranných vývojových PCB, které byly nařezány na rozměry krytky, následně spojeny pomocí dvou M3 šroubů a matice. Takto vytvořená deska byla osazena dvěma 6P4C konektory (více známé jako RJ11/RJ14).

Dále byla osazena patřičnými GPIO konektory, které byly propojeny s konektorem určeným pro mikrokontrolér pro I²C komunikaci podle výše uvedeného schématu zapojení. Dva samčí piny na desce byly určeny pro externí senzory, z nichž jeden je neobsazený.

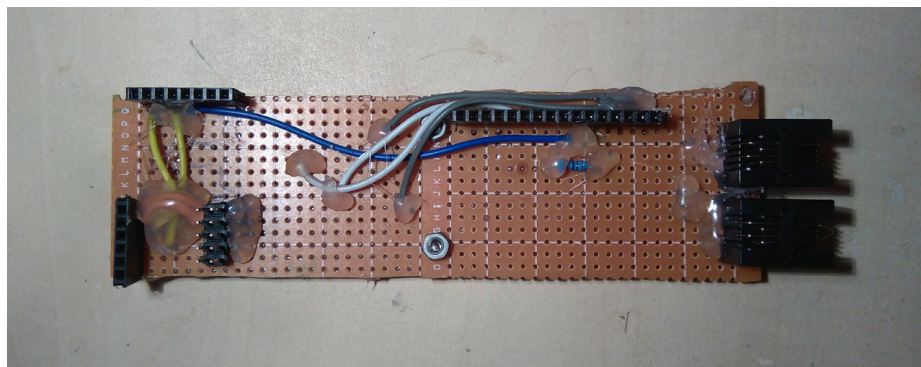
Všechny vodiče byly zality lepidlem z tavné pistole, zejména kvůli ochraně před případným zkratem nebo mechanickým poškozením. Tavné lepidlo bylo vybráno z důvodu snadného odstranění v případě nutnosti úpravy desky (izopropylalkohol i při mírném potření okolí spoje způsobí, že lepidlo jde snadno odlepit, aniž by se poškodily spájené spoje).

Poté byl k desce přišroubován držák, který byl vytisknut na 3D tiskárně (6.4) a slouží k umístění desky včetně krytky na meteostanici.

Do GPIO slotů byly vloženy AS3935 (senzor blesků), BMP280 (senzor teploty a tlaku) a Wemos ESP-WROOM-02. K GPIO samčím konektorům byla připojena telefonní 4 linka, z nichž pak vedou vodiče do senzorů TSL2561 (senzor intenzity světla) a VEML6070 (senzor intenzity UV záření). Tyto senzory se nachází v plastové krabici, která chrání tato zařízení



Obrázek 6.2: Schéma zapojení desky meteostanice s použitím Wemos ESP-WROOM-02



Obrázek 6.3: Neosazená deska

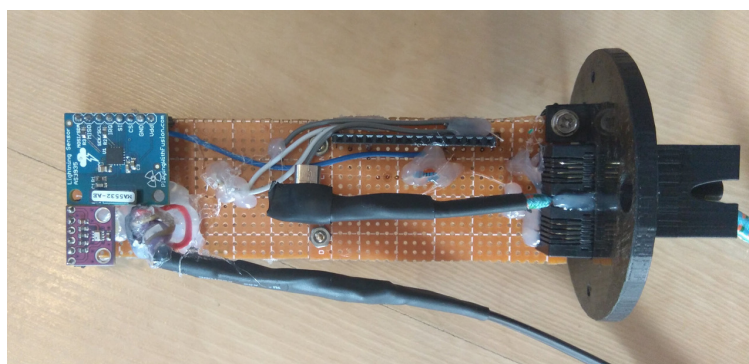
před deštěm a vlhkostí, senzory jsou uvnitř zataveny tavným lepidlem a je uvnitř umístěn sáček se sílicí gelem (pohlcuje vlhkost).



Obrázek 6.4: Senzory světla a uv záření

Zařízení je potřeba i napájet a nabíjet baterii uvnitř, a tudíž bylo potřeba přidat napájecí kabel s microUSB konektorem. Jelikož je ale uvnitř krytu velmi málo prostoru, tak standardní kabel nebylo možné použít. Při měření vyšlo najevo, že ani existující pravoúhlé kabely není možné pro tento způsob využít a bylo potřeba si vyrobit kabel vlastní. K tomuto posloužil levný plochý microUSB kabel, u kterého se odstranil původní konektor a byl připájen nový, otočený do pravého úhlu. Nově vzniklý konektor byl zaizolován smršťovací bužírkou a díky ní a vhodnému umístění vodičů se podařilo vyrobit plnohodnotný kabel.

Kabel byl veden malým prostorem mezi 6P4C konektory a následně zajištěn tavným lepidlem stejně jako konektory k externím senzorům a všechny vodivé části senzorů na desce.

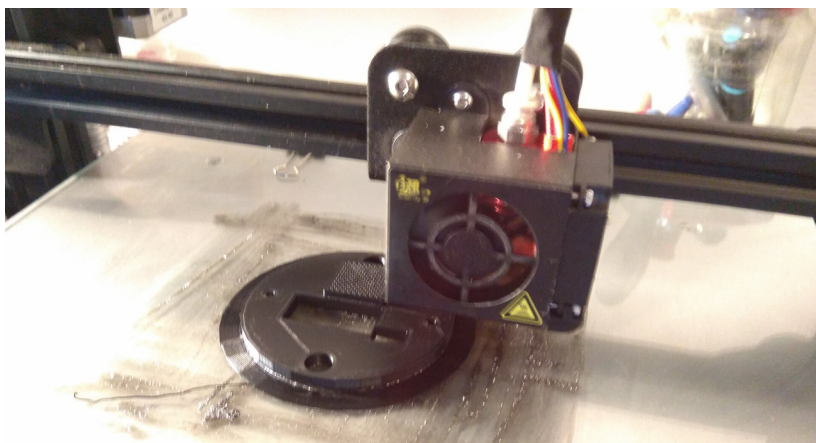


Obrázek 6.5: Deska osazená senzory včetně vyrobeného microUSB kabelu a kabelu k externím senzorům

6.4 3D tisk držáku desky v meteostanici

Jelikož KIT pro meteostanici SEN-08942 neobsahoval držák na desku s MCU a senzory, ale pouze jejich kryt, bylo potřeba vytisknout si tento díl na 3D tiskárně. Tento díl se podařilo nalézt na službě pro sdílení 3D modelů¹ a tudíž tisk byl pouze o správném nastavení tiskárny.

Materiál pro tisk byl zvolen ABS (akrylonitrilbutadienstyren), který je odolný vůči venkovním povětrnostním podmínkám, avšak se s ním špatně tiskne. Naštěstí tiskárna toto zvládla bez problémů.



Obrázek 6.6: 3D disk držáku pro desku

6.5 Umístění na meteostanici

Osazená deska byla umístěna do krytky a zajištěna M3 šrouby. Kabel k externím senzorům (senzor intenzity světla a senzor intenzity uv záření) je vedený ze spodní části krytky otvorem pro větrání.

¹<https://www.tinkercad.com/things/gJMrVZDnPOW>



(a) Umístění do krytky



(b) Detail na krytku včetně šroubů

Obrázek 6.7: Dokončená a osazená deska včetně krytky

Meteostanice byla složena podle návodu² a na ni byly umístěna senzorová jednotka. Externí senzory byly upevněny k horní části meteostanice.

Celé zařízení bylo poté umístěno na střechu, v takové orientaci, kdy úhel 0° na senzoru směru větru korespondoval se severem.

²https://learn.sparkfun.com/tutorials/weather-meter-hookup-guide?_ga=2.197978476.210863752.1524695372-1146594526.1523885261

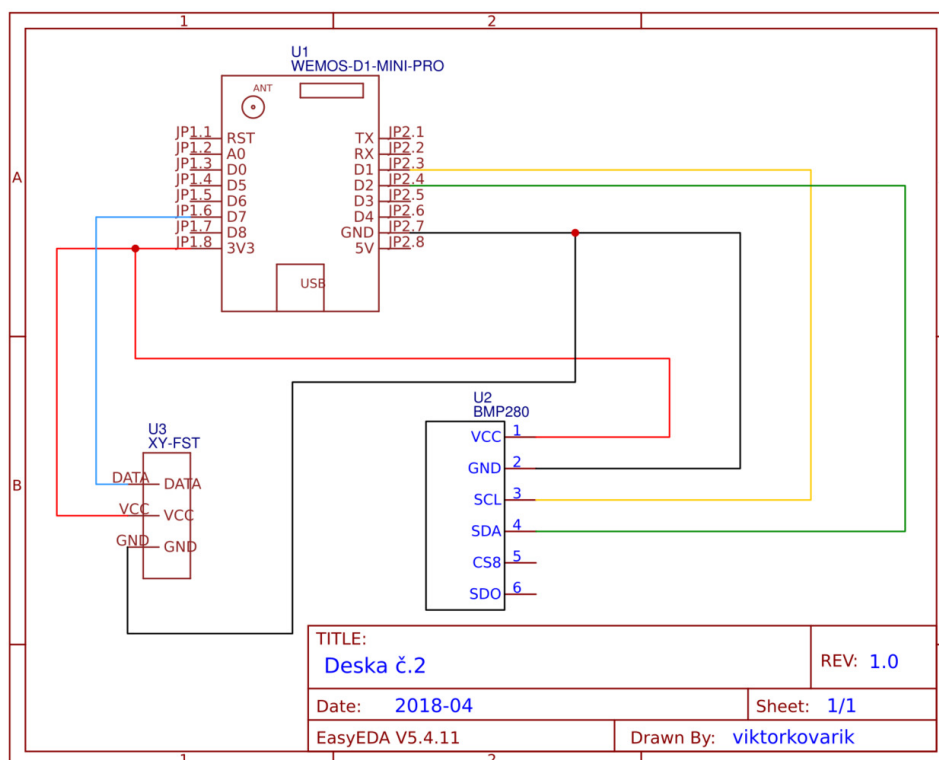


Obrázek 6.8: Dokončená meteostanice na střeše domu

Nabíjecí kabel byl připojen do USB prodlužky a ta husím krkem vedla k nejbližší zásuvce s adaptérem. I přes vzdálenost nabíjení funguje a baterie slouží v případě výpadku napájení, je však zamýšleno použití solárního panelu.

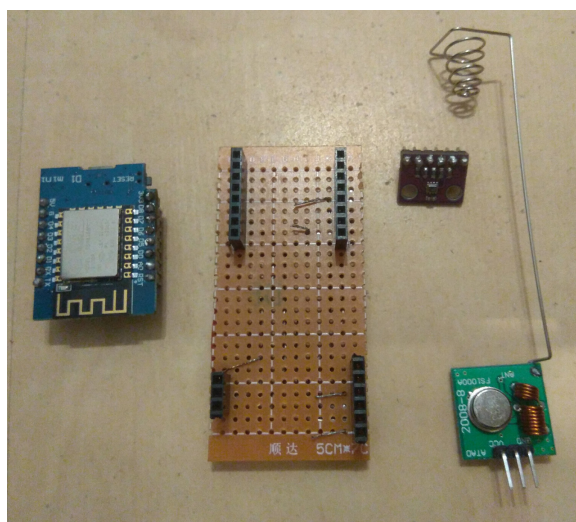
6.6 Výroba desky ovladače žaluzií

Deska pro ovládání žaluzií má jednodušší návrh a obsahuje pouze senzor vnitřní teploty a tlaku BMP280, jehož údaje slouží k lepší automatizaci. Dále je deska osazena 433 MHz vysílačem XY-FST, který byl osazen anténou vlastnoručně vyrobenou, která prodloužila dosah podle měření nejméně na pětinašobek.

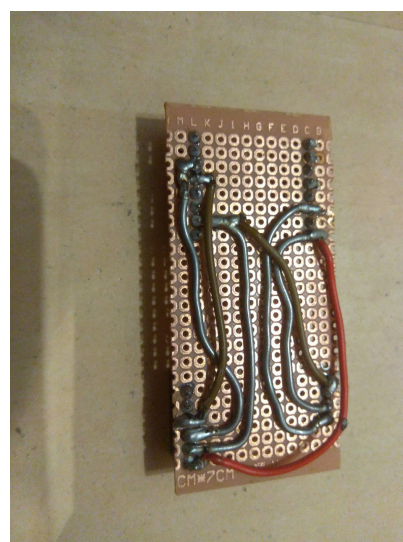


Obrázek 6.9: Schéma zapojení desky s ovladačem žaluzií s použitím Wemos D1 Mini

PCB je podobně jako u venkovní části vyrobena z jednostranné vývojové desky, která byla osazena GPIO sloty, deska obsahuje pouze jeden I²C slot a jeden pro vysílač (napájení a výstup digitálního výstupu, který podporuje PWM).



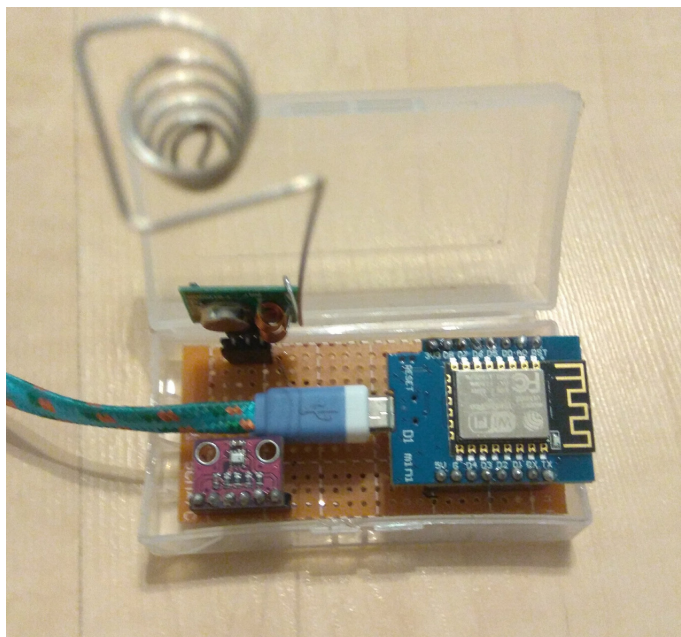
(a) Deska ovládacího modulu pro žaluzie včetně senzoru BMP280 a vysílače XY-FST



(b) Deska ze zadní strany

Obrázek 6.10: Deska pro ovládání žaluzií

Poté byly k desce připojen senzor teploty a vysílač a celá deska byla umístěna do krabičky od 18650 akumulátoru, použitého k napájení meteostanice a následně byla tato deska umístěna v přízemí domu.



Obrázek 6.11: Osazená ovládací deska

6.7 Programování mikrokontrolérů

V následující sekci bude představeno programování meteostanice i ovladače pro žaluzie za využití jednotek ESP8266.

6.7.1 Knihovna WifiManager

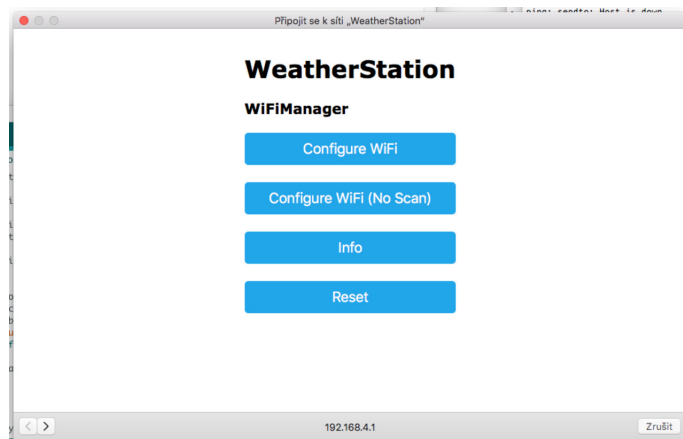
WifiManager je knihovna³ použitá pro usnadnění vytvoření jednoduchého webového rozhraní pro základní konfiguraci jednotky při prvním spuštění.

Při prvním spuštění zařízení, které nemá ve vnitřní paměti uložena data o žádné Wi-Fi síti v okolí dojde k vytvoření Wi-Fi sítě (tzv. hotspot), u které se po připojení zobrazí základní uživatelské rozhraní, pro připojení k Wi-Fi sítím v dosahu.

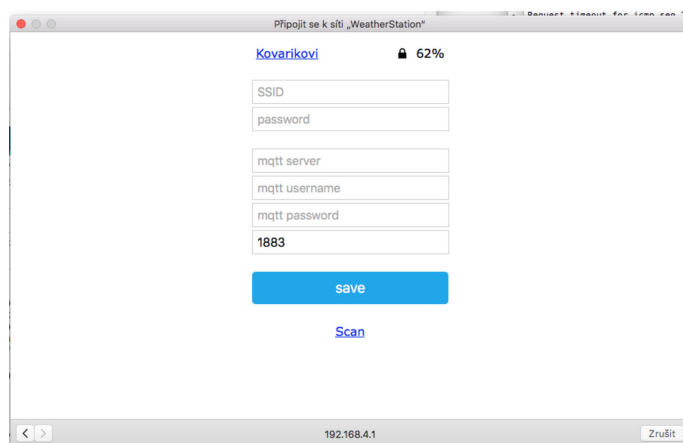
Údaje o síti, vyplněné ve webovém rozhraní je možné poté uložit do vnitřní paměti zařízení a po případném restartu zařízení jsou použity údaje z paměti. Pokud uložené údaje nevyhovují žádné Wi-Fi síti nebo se změnily, je zařízení znovu přepnuto do režimu Wi-Fi hotspotu.

Knihovna také podporuje vlastní formulářová pole, a proto bylo možné přidat pole pro konfiguraci MQTT brokeru a jeho údajů.

³<https://github.com/tzapu/WiFiManager>



Obrázek 6.12: Ukázka rozhraní WifiManageru



Obrázek 6.13: Konfigurace Wi-Fi sítě a MQTT brokeru

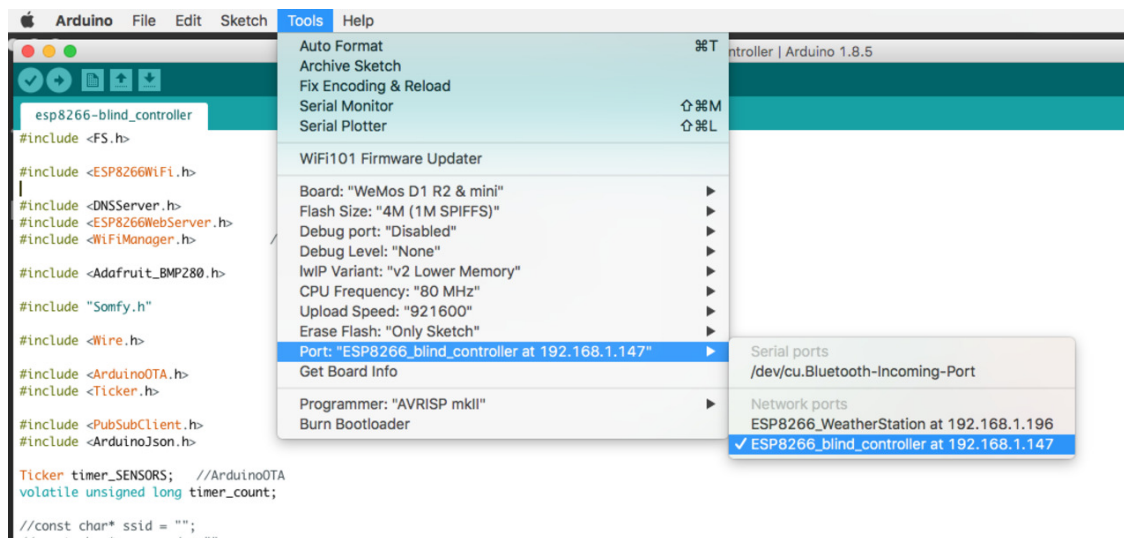
6.7.2 Knihovna ArduinoOTA

Další poměrně důležitou součástí je knihovna ArduinoOTA⁴. Tato knihovna se využívá pro vzdálené nahrání programu do mikrokontroléru skrze Wi-Fi síť.

Tato knihovna funguje tak, že v ArduionoIDE se zobrazí po správném nakonfigurování mezi seriovými COM porty i síťový port zařízení ve Wi-Fi síti. Po jeho zvolení lze využít funkce `Upload sketch` úplně stejně jako to lze přes seriové rozhraní. Rozdíl ovšem je, že u komunikace přes kabel se rovnou nahrává kód do paměti mikrokontroléru, ale u ArduinoOTA funkce se celý kód nahraje nejprve do volné uživatelské paměti a teprve poté, co se dokončí nahrávání dojde k přehrání do programové paměti následované restartem MCU. Poté se už mikrokontrolér spustí s novým kódem.

Nahrávání programů je možné zabezpečit heslem pro upload, pokud není zadáno správné, nenechá se MCU přehrát novým kódem.

⁴<https://github.com/esp8266/Arduino/tree/master/libraries/ArduinoOTA>



Obrázek 6.14: Ukázka podpory pro ArduinoOTA v ArduinoIDE

6.7.3 Programování senzorové jednotky

K inspiraci při implementaci většiny senzorů šlo využít příkladů obsažených v knihovně s programem, avšak AS3935 neměl úplnou knihovnu pro ESP8266. Například kalibrace z knihovny⁵ nebyla funkční, a tudíž byl kód doplněn částí od výrobce použité varianty AS3935⁶, avšak bylo nutné kód přepsat pro ESP8266.

U analogových senzorů se nikde nenacházela žádná knihovna pro ESP8266, tudíž bylo potřeba knihovnu naimplementovat za pomoci dokumentace od výrobce (sekce 3.2).

Popis vlastní knihovny pro SEN-08942

Knihovna se skládá stejně jako ostatní knihovny pro ArduinoIDE z hlavičkového souboru s názvem `SEN_08942.h` a souboru se zdrojovým kódem `SEN_08942.cpp`.

V hlavičkovém souboru se nachází deklarace třídy `SEN_08942`, která je rozdělena na veřejné a privátní metody. Veřejné metody je poté možné volat přímo programem, do kterého je knihovna přidána a složí zejména k předávání dat mezi programem a knihovnou. Privátní metody jsou využity interně pro knihovnu a skrývají před uživatelem pro něj nepotřebné metody.

Na začátku kódu je potřeba deklarovat třídu se kterou potom program bude pracovat a do závorky uvést, na kterých vstupech je anemometr připojený. Například:

```
SEN_08942 anemometer(D3, D4, A0);
```

Kdy první parametr D3 je pull-up pin pro senzor rychlosti větru, D4 je pull-up pin pro senzor srážek a A0 je analogový pin pro senzor směru větru.

Poté je potřeba ve funkci `void setup()` zavolat:

```
anemometer.begin();
```

A ve `void loop()` se potom pro zjištění jednotlivých údajů volají metody:

⁵<https://github.com/bodgit/AS3935>

⁶https://github.com/PlayingWithFusion/PWFusion_AS3935_I2C

```
windspeed = anemometer.get_windspeed();
rainfall = anemometer.get_rainfall();
winddirection = anemometer.get_winddirection_degrees();
```

První metoda slouží k zjištění aktuální rychlosti proudění vzduchu. Druhá slouží ke zjištění počtu milimetrů srážek za poslední hodinu a třetí metoda zjišťuje směr větru ve stupních.

Mimo ně jsou v knihovně metody pro návrat zkratky světové strany místo stupňů a také možnost definovat, kolik stupňů z třetí metody připadá na sever.

V případě, že zařízení neměří přesně, je v knihovně funkce na kalibraci analogového vstupu, standardní Arduino má 1024 hodnot, avšak u některých desek může číst špatně maximální hodnotu a interpretovat ji třeba jako nižší číslo).

Díky použití knihovny `ticker.h` jsou všechny hodnoty počítány průběžně a neblokují hlavní `void loop()` smyčku programu a vrací hodnoty v reálném čase.

6.7.4 Programování ovládání žaluzií

Pro ovládání Somfy zařízení na ESP8266 rovněž neexistovala žádná knihovna a nejpíše ani příklady, proto bylo potřeba si napsat knihovnu svoji. K implementaci bylo využito částečně testovacího python skriptu pro Raspberry Pi z předešlé doby a příkladu pro Arduino⁷ (avšak u tohoto příkladu bylo autorem využito ovládání přes registry MCU a tudíž daný příklad nešel použít).

Při implementaci bylo využito knihovny `Wire.h` a také knihovny pro práci s interní flash pamětí `FS.h`, která slouží zejména k uchování údajů od již použitých ovladačů.

Podobně jako u knihovny pro SEN-08942 je zde několik veřejných metod přístupných z ArduinoIDE a také interní (private) metody pro využití uvnitř vlastního kódu knihovny (například metody, které se starají o přípravu rámce, sekce 4.2, nebo ukládání a čtení údajů o ovladači).

Podobně jako u knihovny v sekci 6.7.3 je potřeba nejprve třídu deklarovat a uvést při tom, na kterém výstupním pinu je připojen datový pin vysílače (sekce 5.1.3) a adresu ovladače, se kterým se bude pracovat.

```
Somfy blind(TRANSMITTER_433MHZ_PIN, address);
blind.begin();
blind.send_command(command);
```

V kódu ovladače žaluzií (sekce 6.6) se tato třída inicializuje při obdržení MQTT zprávy s příkazem, nastaví se správně pin vysílače a adresa se načte do paměti.

Metoda `blind.begin()`, provede nastavení zvoleného pinu na výstup a načte rolling code z paměti MCU (pokud existuje).

Poslední veřejná metoda `blind.send_command(command)` provede odeslání příkazu žaluziím. Nejdříve je vytvořen datový rámec (sekce 4.2), poté je poslán signál na probuzení žaluzií, společně s hardwarovou a softwarovou synchronizací (sekce 4.2), které následuje již vytvořený datový rámec.

6.8 Implementace řídicí jednotky

V této sekci je popsána implementace řídicí jednotky a automatizace.

⁷https://github.com/Nickduino/Somfy_Remote

6.8.1 MQTT broker

Na stejném serveru, jako se nachází řídicí jednotka byl nainstalován broker server Mosquitto. Tento broker se velmi jednoduše instaluje i spravuje. MQTT broker má na starost vystupovat jako centrální bod pro MQTT protokol a má za úkol směřovat zprávy, které se pomocí tohoto protokolu přenášejí.

6.8.2 Automatizační program

Python aplikace, která automatizuje chování žaluzií je vlastně MQTT klient, který odebírá všechny témata, do kterých publikují senzory po domě a doručená data dále zpracovává a rozhoduje posléze o akci.

Program má definované jednotlivé žaluzie a také skupiny, do kterých žaluzie patří podle světových stran. Toto nastavení umožňuje ovládat buď jednotlivé žaluzie, všechny naráz nebo například pouze žaluzie na západní, či jiné straně domu.

V případě, že připojení k brokerovi je úspěšné, dochází k zahájení `while True` cyklu, kde se data získávají z MQTT témat a podle jejich typu ukládají do třídy `State`, která obsahuje pro každý senzor patřičnou proměnou a často také zásobníky několika předešle naměřených hodnot (například posledních několik hodnot teploty venku pro zjištění průměrné teploty za poslední dobu nebo zásobník hodnot směrů větrů pro zjištění převažujícího směru větru během poslední doby).

S těmito údaji poté nakládají jednotlivá automatizační pravidla (sekce 5.2.2), která jsou napsaná jako třídy implementující rozhraní `AutomationRule`:

```
class AutomationRule:
    def checkConditions(self, state):
        pass
    def getPriority(self):
        pass
    def getTargetPosition(self):
        pass
    def getTargetBlind(self):
        pass

class RainPrevention(AutomationRule):
    rain = 8 # strong rain
    def checkConditions(self, state):
        if state.rainfall is None: return
        return state.rainfall > self.rain
    def getPriority(self):
        return 6
    def getTargetPosition(self):
        return Position.HALF_OPEN
    def getTargetBlind(self):
        return blind_all
```

Výše uvedené pravidlo je nejjednodušší v kódu, protože o průměrný počet mm srážek za hodinu se stará již mikrokontrolér a tudíž není potřeba řešit žádné zásobníky a časovače. Jednotlivá pravidla v kódu se povolují přidáním instance třídy daného pravidla do seznamu

pravidel, například: `activeRules.append(HeatPreventionRule())`. Tím pádem se dají pravidla, která si nepřejeme momentálně využívat, dočasně vypnout.

Každé pravidlo má svoji prioritu, přičemž pravidla, která kontrolují ovládání žaluzií na základě denní doby jsou platná téměř vždy, ale mají nejnižší prioritu. Například bouřka má jednu z nejvyšších priorit, co se týče pravidel ovládání žaluzií. se žaluzie daly do patřičného stavu. Tím pádem, jakmile bouřka (která začala ve dne) skončí ve večerních hodinách, začne platit pravidlo nočního zatažení žaluzií a již se nevrací žaluzie do stavu před bouřkou (roztaheno ve dne).

Je zde například pravidlo pro silný vítr, který však není silný natolik, že by byl hrozbou pro všechna okna. Toto pravidlo, má definováno, aby se v případě jeho platnosti provedlo uzavření jen určité skupiny žaluzií. Toho je docíleno seznamy, které obsahují jednotlivé adresy žaluzií podle světových stran.

6.9 Ověřování validity senzorů

• Meteostanice

Pro účely debugingu byla vyrobena jednoduchá deska, která pouze vodiče z RJ14 slotu na konektory GPIO. Byla vyrobena zejména z důvodu snadnější přenositelnosti a případné manipulovatelnosti.

Deska pro svoji univerzálnost byla využita pro všechny senzory z meteostanice. Prostřední vodiče slouží vždy pro senzory větru a deště, a krajní vodiče z každé strany slouží k připojení potenciometru ze senzoru směru větru. Poslední část desky je pátý vodič, který je podobně jako na desce meteostanice je vybaven 10 kOhm rezistorem a slouží jako vstupní napětí k děličce.

Dále byl vyroben kabel RJ14, který na své druhé straně měl 4 GPIO výstupy. Tento kabel byl použit zejména pro otestování vodivosti jednotlivých slotů jak na meteostanici, tak později u testovací desky.

K testování byl využit Wemos D1 Mini z ovladače žaluzií z toho důvodu, že se snáze demontuje. Testování validity těchto senzorů vždy bylo vybíráno ve vhodné době z hlediska automatizace, aby dočasná nepřítomnost tohoto zařízení tolik nevadila.

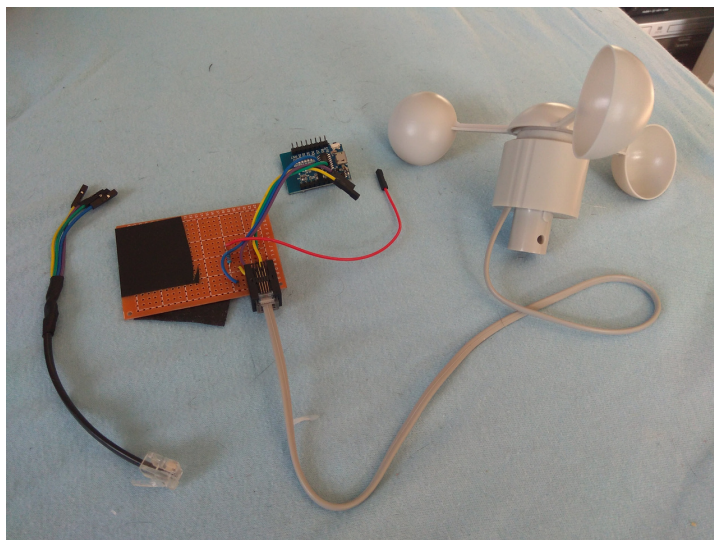
• I²C senzory

Tyto senzory byly porovnávány většinou s naměřenými údaji jiných zařízení nebo porovnáním s reálnou situací.

6.9.1 Senzor rychlosti větru

Testování senzoru rychlosti větru probíhalo při bezvětřném dnu a to prostým rozjetím automobilu na rychlost cca 50 km/h a měřila se naměřená data.

Z obrázku 6.16 vyplývá, že v dokumentaci k senzoru je vzorec pro výpočet rychlosti uveden správně. Tachometr v autě ukazuje rychlost 50 km/h a naměřená hodnota byla 48.80 Km/h. Menší rozdíl v rychlosti lze přisuzovat zejména tomu, že tachometry v autech často ukazují trošku více a také možnému působení slabého větru.



Obrázek 6.15: Testovací deska, Wemos D1 Mini, RJ14 kabel pro debugging

6.9.2 Senzor směru větru

Testování senzoru pro směr větru bylo celkem triviální, byly změřeny odpory na jednotlivých pozicích senzoru směru větru a porovnány s dokumentací [12]. Tato data odpovídala tabulce, lišila se maximálně o pár ohmů.

Problém trochu nastal u A/D převodníku. V dokumentaci byly uvedeny hodnoty napětí pro 5 V, což odpovídá analogovému vstupu Arduina (sekce 3.3.1). Bylo tedy nutné přeměřit a přepočítat hodnoty z dokumentace na 3.3 V. Poté, jelikož analogové senzory mohou měřit nepřesně, tak předejít chybám byl kód pro rozpoznávání směru větru napsán tak, aby vybral nejbližší hodnotu z referenčních a podle toho vrátil směr větru ve stupních.

V případě, že se hodnoty blížily hodnotě 0 (analogový vstup bez vstupního napětí) nebo 1024 (na analogovém vstupu je napětí 3.3 V), vrací knihovna úhel 720°, který lze pak jednoduše ošetřit v kódu.

6.9.3 Senzor srážek

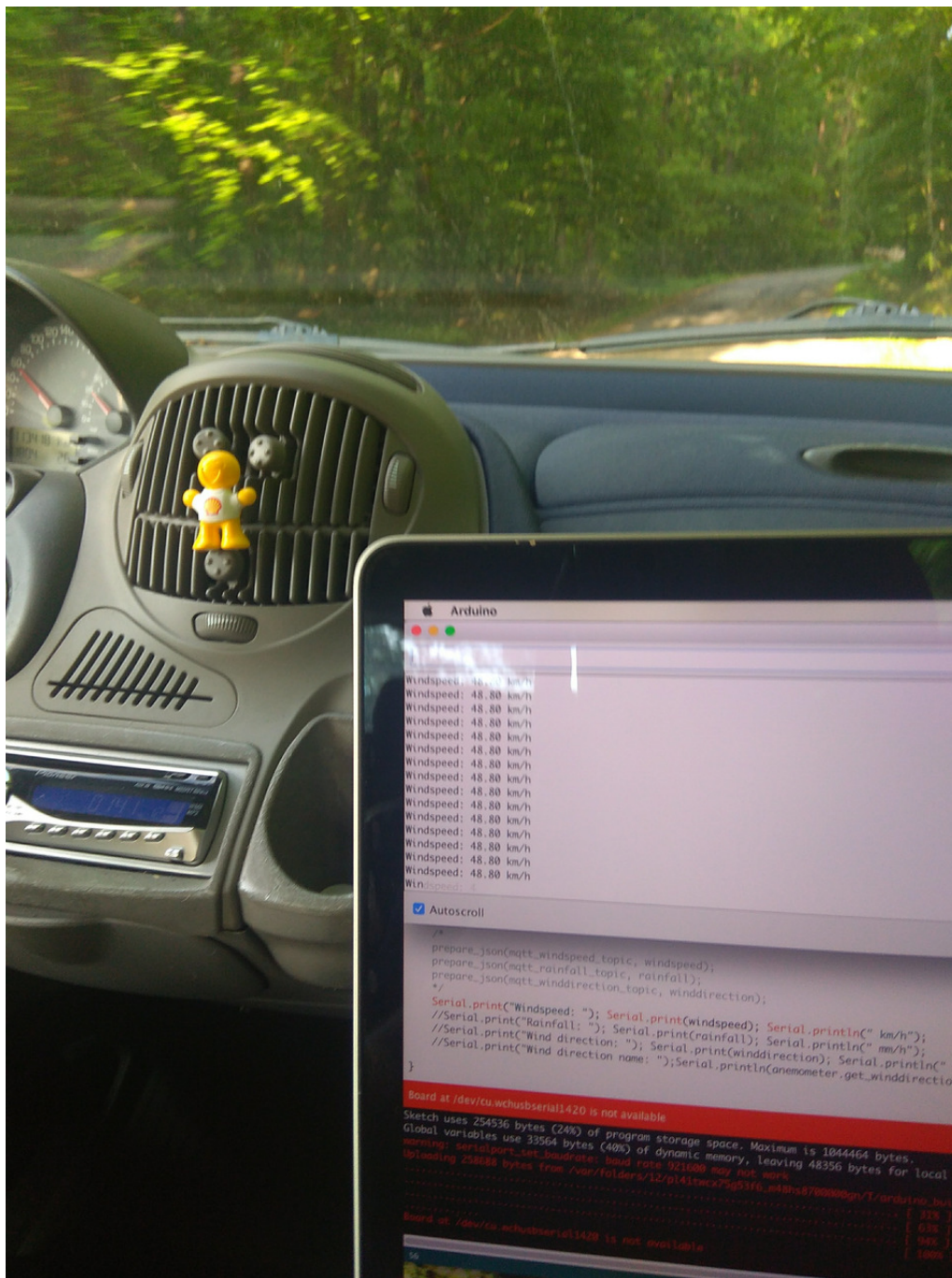
Ověření validity senzoru srážek spočívalo v nalití 100 ml vody do vaničky senzoru srážek a po protečení vody byla zaznamenána naměřená hodnota mikrokontrolérem.

Při prvním nalití 100 ml vody bylo naměřeno 10.8966 mm srážek za hodinu. Vanička senzoru deště má rozměry 11x5 cm (0.0055 m²). Když se tyto naměřené údaje porovnají s údaji z kalkulačky⁸, vyjde, že na 10.5899 mm srážek za hodinu je potřeba zhruba 60 ml vody. Tento údaj však nesedí se 100 ml vody, které byly vlitý do senzoru srážek.

Proto byl pokus zopakován při mnohem pomalejším lití vody do vaničky, kdy byla naměřena hodnota 17.6022 mm srážek za hodinu, po dosazení do kalkulačky toto odpovídalo 96.8 ml nalité vody do senzoru srážek.

Z těchto údajů vyplývá, že senzor srážek po potřeby automatizace žaluzií by měl stačit. Příliš velké množství srážek za krátkou dobu (jednotky milimetrů za vteřinu, stovky ml vody za méně než minutu) by mohly způsobit zkreslené hodnoty.

⁸<http://www.calctool.org/CALC/other/default/rainfall>

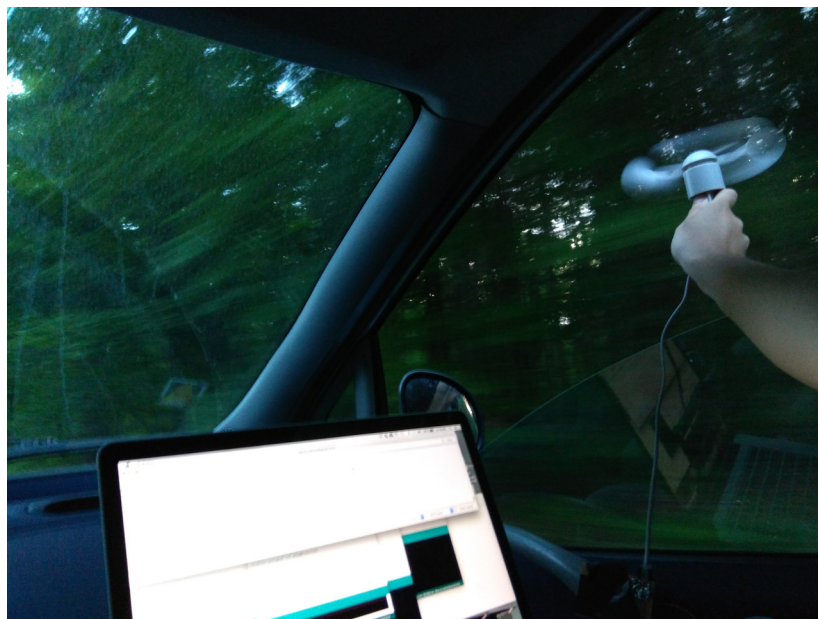


Obrázek 6.16: Testování v autě při bezvětří

6.9.4 BMP280

Při ověřování přesnosti senzoru teploty byla teplota porovnávána s několika teploměry, dvěma rtuťovými a jedním digitálním. Všechny teploměry byly ve stínu.

Při měření bylo zjištěno, že digitální teploměr je velmi nepřesný, měří asi o 8 stupňů vyšší teplotu i ve stínu, proto byl z testu později vyřazen. Oba senzory BMP280 měřily téměř



Obrázek 6.17: Pohled ven

identicky, rozdíl byl později patrný pouze u BMP280 umístěného v krytce, kdy v bezvětrí a slunečném dni byly občas teploty o cca 2 °C vyšší, než v naměřené jiným teploměrem, toto se projevilo patrně přítomností mikrokontroléru v krytce a jím generované teplo pokud vítr nefoukal velmi mírně zvýšilo naměřenou teplotu.

6.9.5 TSL2561 a VEML6070

Testování obou senzorů probíhalo na základě porovnání s reálnými venkovními podmínkami.

U senzoru okolního světla (TSL2561) probíhalo testování sledováním chování obyvatel domu, kteří v určitou denní dobu a při určitém venkovním osvětlení ručně zatahovali žaluzie, tyto hodnoty odpovídaly průměrně 13 luxům, přičemž jedna z testovaných osob dávala často zatahovala žaluzie i při mnohem vyšších hodnotách osvětlení, avšak druhé osobě to nevyhovovalo. Testované osoby o testu nevěděly, aby neměly ovlivněný úsudek.

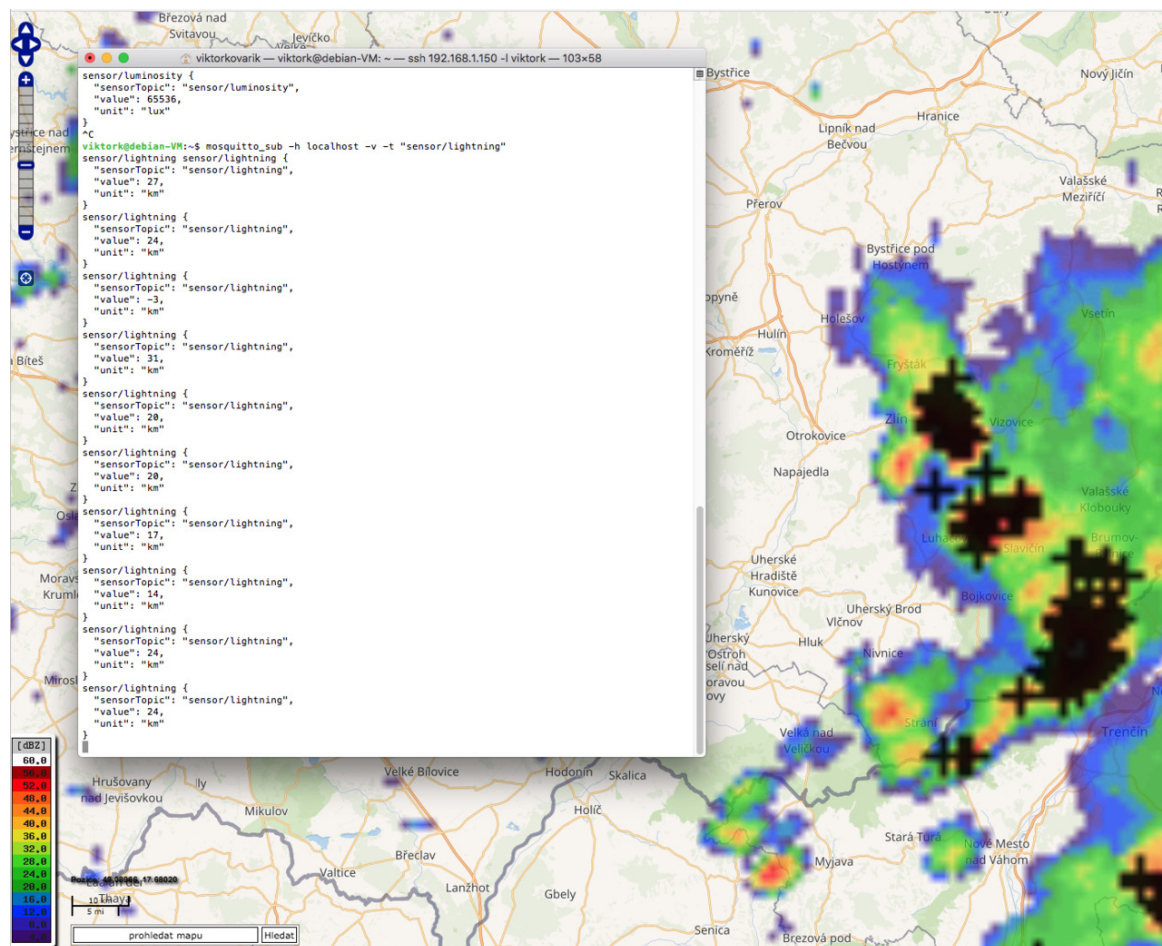
Senzor uv záření byl řádně otestován díky nezvykle dobrému počasí na během dubna roku 2018, kdy počasí bylo velmi podobné letnímu. Naměřené hodnoty v nejvíc horké dny se blížily hodnotám mezi 1300–1500 jednotek (tyto hodnoty odpovídají dle dokumentace k senzoru [11] hodnotám uv indexu 6–7, což je vysoká hodnota). Proto jako hranice uzavření oken do půlky z důvodu horka a velké sluneční aktivity byla stanovena hodnota vyšší než 1500 jednotek (odpovídá velmi vysokému uv záření).

6.9.6 Senzor blesků AS3935

Testování tohoto senzoru bylo nejtěžší ze všech, od pořízení toho senzoru, mezi říjnem 2017 až dubnem 2018 nebyla žádná bouřka v jeho dosahu zaznamenána⁹. Naštěstí v závěru psaní práce byla kratší bouřková přehánka zaznamenána a zdokumentována.

⁹http://report.bourky.cz/?page=report_list

Senzor umožňuje kalibraci, špatná kalibrace se projevuje zejména zprávami typu „disturber“ (což je například vysokofrekvenční signál vyvolaný nastartovaným motorem nebo například i datovým přenosem I²C poblíž antény na senzoru). Výrobce senzoru naštěstí, ale každý senzor, který prodává kalibruje a posílá zákazníkům hodnotu pro využití v kalibrační funkci programu (u dodaného senzoru tato hodnota odpovídá kapacitě 72 pF). Díky této kalibraci pak senzor neposílá nesprávné údaje vyplývající z různého rušení a tím pádem nedostávaly žaluzie nesprávné příkazy na uzavření při bouři.



Obrázek 6.18: Bouřka dne 12. 5. 2018. Pozn. hodnota „-3“ v datech znamená naměřenou bouřku mimo dosah senzoru (> 40 km)

Vzhledem k problémům s nedostatkem bouřek (a tím pádem dat) v zimním období by se tedy na naměřenou vzdálenost od bouřky nedá příliš spoléhat, nicméně bouřka byla detekována a žaluzie nastaveny, aby se zavření pouze, pokud je vzdálenost menší jak 10 kilometrů.

Kapitola 7

Nasazení a testování

Po dokončení implementace byl systém nasazen a otestován. V následující kapitole je vysvětlen tento průběh.

7.1 Instalace systému

- **Zprovoznění mikrokontrolérů**

Instalace systému do mikrokontrolérů je nejjednodušší pomocí ArduinoIDE¹, do složky s ostatními knihovnami je potřeba nakopírovat knihovnu pro SEN-08942 (sekce 6.7.3). Dále také upravenou knihovnu pro AS3935², do které byly přidány nějaké funkce z knihovny výrobce použitého senzoru (první knihovně chyběla funkční kalibrace a druhá není kompatibilní s ESP8266). V neposlední řadě bylo potřeba do složky s knihovnami nakopírovat knihovnu pro ovládání žaluzií Somfy (sekce 6.7.4).

Poté je potřeba doinstalovat knihovny pro použité senzory (sekce 3.2) a také knihovnu ArduinoOTA (sekce 6.7.2, WiFiManager (sekce 6.7.1) a MQTT klienta³ a Ticker⁴ (časovač, používá se v kódu pro neblokované zpoždění) pomocí ArduinoIDE (Sketch–Include Library–Manage Libraries).

Po zkompileování a nahrání kódu se objeví nové Wi-Fi síť s názvy WeatherStation a BlindController (sekce 6.7.1). Po připojení stačí zvolit domácí Wi-Fi síť, ke které se má zařízení připojit a vyplnit adresu a přihlašovací údaje k MQTT brokeru (sekce 6.8.1).

Zařízení uloží tyto údaje do své vnitřní paměti a restartuje se, po restartu se již posílají data brokeru a také je možnost kód přehrávat i po síti přes ArduinoOTA.

- **Zprovoznění řídicí jednotky**

Ke zprovoznění systému na routeru Turris je vhodné zprovoznit LXC kontejner založený na Debian Linuxu [4]. To zejména z toho důvodu, že systém řízení žaluzií je pak oddělený od zbytku systému routeru a jde deaktivovat jednoduše například i z webového rozhraní.

¹<https://www.arduino.cc/en/Main/Software>

²<https://github.com/bodgit/AS3935>

³<https://github.com/knolleary/pubsubclient>

⁴<https://github.com/esp8266/Arduino/tree/master/libraries/Ticker>

Nicméně veškeré požadavky na systém je mít nainstalovaný libovolný MQTT broker, Python 2.7 nebo Python 3.6 a několik knihoven. Mimo systémové a standardní knihovny jako jsou `sys`, `math`, `cmath` a `time`, `json` (od verze 2.6 součástí instalace Pythonu) je také potřeba doinstalovat knihovny `paho.mqtt.client`⁵ a `ephem`⁶ (slouží k výpočtu západu slunce, pokud nejsou k dispozici data ze senzoru osvětlení).

Po instalaci balíčků je potřeba nastavit v Python skriptu údaje k MQTT brokeru, adresy žaluzií a také aktuální souřadnice pro ovládání v případě nefunkčnosti senzoru intenzity světla.

- **Přidání ovladačů do žaluzií**

Prvotní párování je potřeba provést za pomoci sekce 4.2 a libovolného MQTT klienta (broker servery obsahují i klienty), příkaz pro párování vypadá například následovně:

```
mosquitto_pub -t "action/blind"-m '{"address": 0x121310, "command": 0x8}'
```

Kdy položka `address` obsahuje adresu, kterou chceme přidat a položka `command` obsahuje příkaz pro žaluzie (v tomto případě tlačítko *PROG*).

7.2 Testování funkčnosti systému

Nejprve bylo potřeba otestovat, zda systém nebo jeho části jsou stabilní. V tomto případě tedy, zda mikrokontroléry v průběhu dne i různých povětrnostních podmínek posílají validní data a zda řídicí jednotka na tato data správně reaguje.

Během testování panovala jak velká horka (30°C), tak silné deštové přeháňky (cca 40 mm/h). Tyto situace jednotka na střeše zvládla a celou dobu posílala data.

Testování automatizace probíhalo tak, že byl program nejdříve spuštěn bez přístupu k ovládání žaluzií a byly sledovány celý den jeho provedené příkazy.

Hned ze začátku byla přidána průměrná hodnota téměř⁷ všech naměřených údajů za posledních pět minut. Průměrné hodnoty zabránily například chvilkově zvýšené rychlosti větru nebo například oblačnosti při západu slunce, kdy může chvilkově klesnout svit slunce na hranici západu a pak se zase zvýšit.

Dále u pravidel pro déšť, bouřku, rychlý vítr a velký svit slunce byla zavedena prodleva. Pravidlo, které se aktivuje, platí určitou dobu, například nejméně 20 minut od poslední události. Od senzoru blesků totiž by se žaluzie se neustále uzavíraly a otevíraly při každé jednotlivé události a například u pravidla pro vysokou teplotu a sluneční svit (sekce 5.2.2) docházelo při přeletu menší oblačnosti, či jen nějakého stínu k opakujícím se příkazům (obrázek 7.1), což opotřebovává motor.

⁵<https://pypi.org/project/paho-mqtt/>

⁶<https://pypi.org/project/pyephem/>

⁷U senzoru bouřky se nepočítá průměrná vzdálenost, jelikož blesky byly zaznamenávány jak daleko, tak blízko, obrázek 6.18 a senzor srážek vrací hodnoty v milimetrech za hodinu.

```

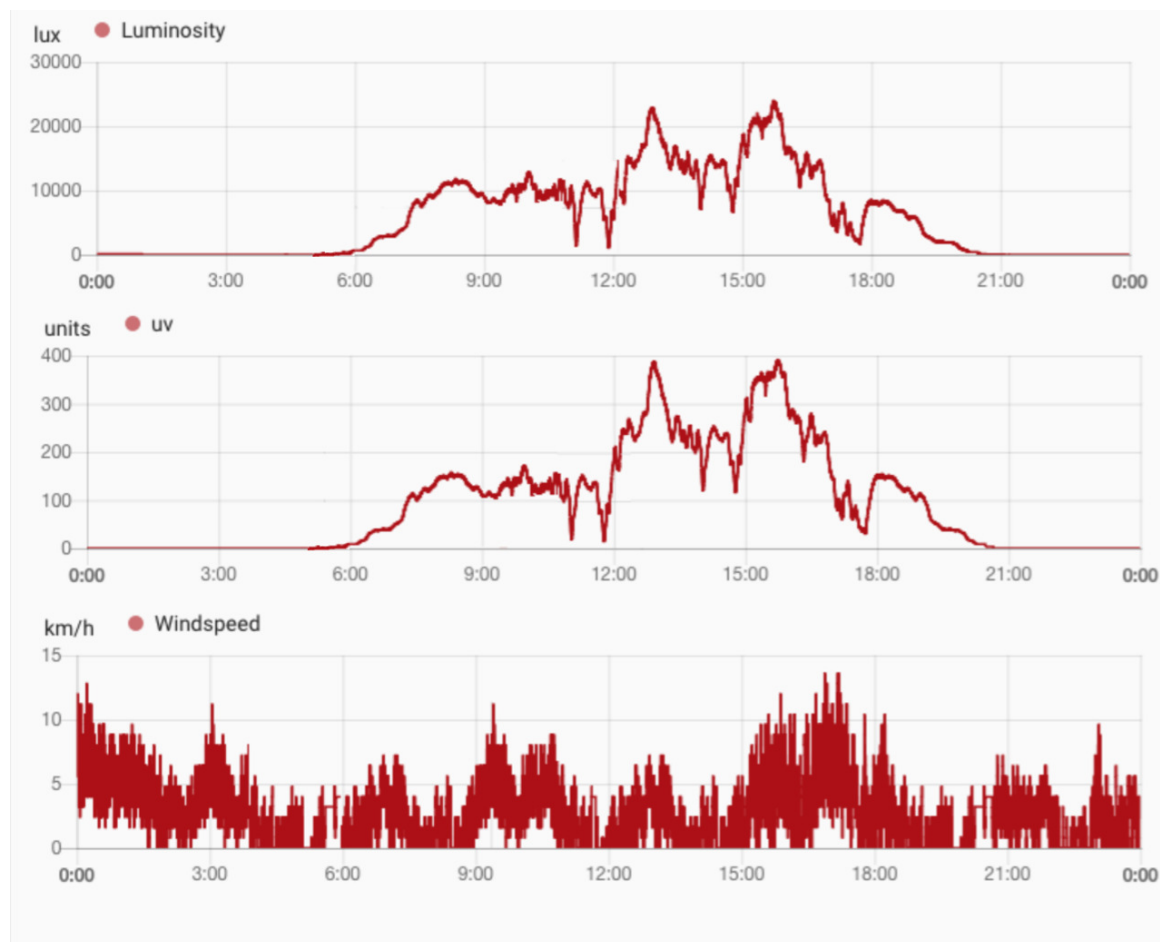
viktork@debian-VM:~$ python3 ibt-blind-driver.py
Rule: NoLumiSensorRule, new position: Position.OPEN, blind: 1184528, time: 11:36, 2018-05-12
Rule: DayOpenRule, new position: Position.OPEN, blind: 1184528, time: 11:36, 2018-05-12
Rule: HeatPreventionRule, new position: Position.HALF_OPEN, blind: 1184528, time: 12:50, 2018-05-12
Rule: DayOpenRule, new position: Position.OPEN, blind: 1184528, time: 12:58, 2018-05-12

```

Obrázek 7.1: Na obrázku lze spatřit chybovou situaci, kdy při horkém dni nachvíli došlo k poklesu UV záření a během pár minut se žaluzie vrátily do původního stavu

7.3 Výsledky z reálného používání

Následující grafy zobrazují zaznamenaná data během oblačného dne.



Obrázek 7.2: Grafy vývoje slunečního světla, uv záření a rychlosti větru během oblačného dne (15.5.2018)

Z grafu vyplývá, že slunce vycházelo krátce před pátou hodinou ránní a zapadalo chvíli před devátou hodinou večer⁸, což lze pozorovat i na výpisu z programu pro automatizaci (obrázek 7.3).

⁸<https://www.meteogram.cz/vychod-zapad-slunce/>

```

viktork@debian-VM:~$ python3 ibt-blind-driver.py
Rule: NoLumiSensorRule, new position: Position.OPEN, blind: 1184528, time: 18:44, 2018-05-13
Rule: None, new position: Position.OPEN, blind: None, time: 18:45, 2018-05-13
Rule: NightCloseRule, new position: Position.CLOSED, blind: 1184528, time: 20:48, 2018-05-13
Rule: None, new position: Position.CLOSED, blind: None, time: 04:46, 2018-05-14
Rule: DayOpenRule, new position: Position.OPEN, blind: 1184528, time: 05:00, 2018-05-14
Rule: None, new position: Position.OPEN, blind: None, time: 18:00, 2018-05-14
Rule: NightCloseRule, new position: Position.CLOSED, blind: 1184528, time: 20:26, 2018-05-14
Rule: None, new position: Position.CLOSED, blind: None, time: 04:50, 2018-05-15
Rule: DayOpenRule, new position: Position.OPEN, blind: 1184528, time: 05:00, 2018-05-15
Rule: None, new position: Position.OPEN, blind: None, time: 18:00, 2018-05-15
Rule: NightCloseRule, new position: Position.CLOSED, blind: 1184528, time: 20:49, 2018-05-15
Rule: None, new position: Position.CLOSED, blind: None, time: 04:41, 2018-05-16
Rule: DayOpenRule, new position: Position.OPEN, blind: 1184528, time: 05:00, 2018-05-16

```

Obrázek 7.3: Automatizované chování během několika dnů nepřetržitého provozu

Na obrázku 7.3 si dále můžeme povšimnout, že každý den je čas východu slunce přesně v 5:00, toto je způsobeno pravidlem, které stanovuje, aby vysouvání žaluzií probíhalo pouze v denní dobu (a tím se zabránilo příliš brzkému vytažení žaluzií nebo například chybě senzoru).

Tím pádem by systém měl fungovat správně.

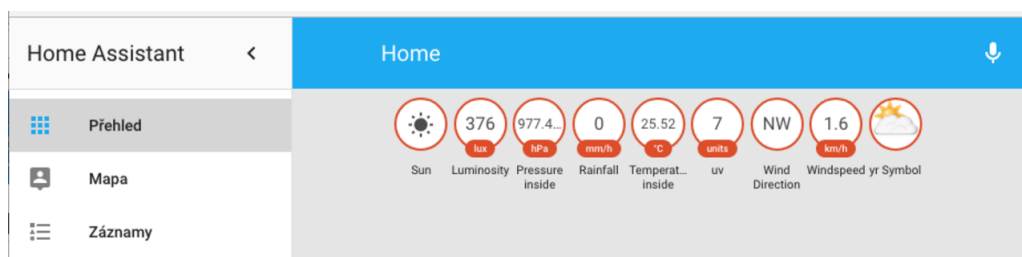
Kapitola 8

Závěr

V práci bylo zpracováno automatické ovládání žaluzií Somfy na základě dat ze senzorů. Práce i přes všechny různé hardwarové (nefunkční BMP280, poškozený mikrokontrolér, komplikované pájení) i softwarové (stabilita systému, chyby v chování, `return` v `loop()` funkci mikrokontroléru) komplikace došla do stádia, kdy systém je plně autonomní a běží již několik týdnů bez výraznějších zásahů z venčí.

Projekt díky použití MQTT protokolu pro komunikaci, jazyku Python pro automatizaci a mikrokontrolérů založených na ESP8266 je velmi univerzální a nezávislý na platformě a prostředí. Z hlediska rozšiřitelnosti je možné například meteostanici rozšířit novými senzory (bez potřeby pájení jsou k dispozici na desce rovnou konektory pro další I²C zařízení 6.3) a nebo stávající řešení napojit například na Google Home¹, projekt BeeeOn² nebo Home Assistant³ a také data ze senzorů poté distribuovat do přehledných aplikací pro mobilní telefony nebo jako webové stránky.

Propojení s home-assistant.io již funguje (obrázky 8.1, 8.2) a v průběhu léta bude patrně realizováno propojení s Google Home, po domě jsou k dispozici různě rozmístěná Raspberry Pi (fungující jako multimediální centra), která mohou sloužit jako asistenti. Využilo by se patrně služby IFTTT⁴, která umožňuje jednoduchými pravidly definovat různá chování. Jako další nápad by mohlo být pomocí této služby realizovat varování před bouřkami.



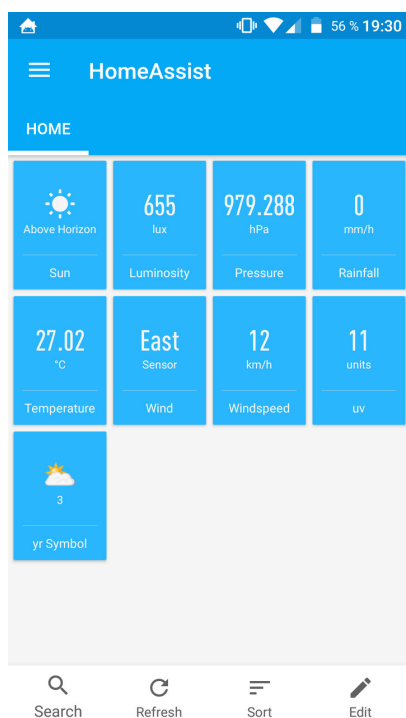
Obrázek 8.1: Funkční napojení stávajícího projektu na home-assistant.io

¹https://store.google.com/product/google_home

²https://beeeon.org/wiki/Main_Page

³<https://www.home-assistant.io/>

⁴<https://ifttt.com>



(a) Přehled senzorů



(b) Graf vývoje intenzity uv záření při deštivém dni

Obrázek 8.2: Webová verze home-assistant.io

Další možností rozšíření projektu je například propojení stávajícího systému se systémem vytápění v domě za využití stávajícího plynového kotle, údajů o venkovní i vnitřní teplotě a například také zapojení určitého plánování vytápění, aby v domě v dobu návratu obyvatel z práce či školy bylo již zatopeno. To by jednou mohlo být předmětem nějaké z dalších prací.

Literatura

- [1] AMS: *AS3935 Franklin Lightning Sensor IC*. [Online; navštíveno 12.05.2018].
URL https://www.embeddedadventures.com/datasheets/AS3935_Datasheet_EN_v2.pdf
- [2] Bosch: *BMP280 Digital Pressure Sensor*. [Online; navštíveno 12.05.2018].
URL <https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>
- [3] CZ.NIC: *Projekt Turris*. [Online; navštíveno 13.05.2018].
URL <https://project.turris.cz/cs/>
- [4] CZ.NIC: *Projekt Turris - Linuxové kontejnery*. [Online; navštíveno 13.05.2018].
URL <https://doc.turris.cz/doc/cs/howto/lxc>
- [5] Duval, J.; Neuman, S.; Fournet, M.: *Communication system with cross-compatibility and associated communication frame*. Květen 2012.
URL <https://patents.google.com/patent/US8189620/>
- [6] Inc., T. A. O. S.: *TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER*. [Online; navštíveno 12.05.2018].
URL <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf>
- [7] Leens, F.: An introduction to I²C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, ročník 12, č. 1, 2009: s. 8–13.
- [8] Patel, K. K.; Patoliya, J.; Patel, H.: Low cost home automation with ESP8266 and lightweight protocol MQTT. *Transactions on Engineering and Sciences*, ročník 3, č. 6, 2015: s. 2347–1875.
- [9] Pushstack: *Somfy RTS Protocol*. [Online; navštíveno 10.05.2018].
URL <http://pushstack.wordpress.com/somfy-rts-protocol/>
- [10] Richardson, M.; Wallace, S.: *Getting started with Raspberry Pi*. "O'Reilly Media, Inc.", 2012.
- [11] Semiconductors, V.: *UVA Light Sensor with I2C Interface*. [Online; navštíveno 12.05.2018].
URL <https://www.vishay.com/docs/84277/veml6070.pdf>
- [12] Sparkfun: *SEN-08942 weather sensor*. [Online; navštíveno 12.05.2018].
URL https://cdn.sparkfun.com/assets/8/4/c/d/6/Weather_Sensor_Assembly_Updated.pdf

Příloha A

Obsah přiloženého CD

```
|---- README.TXT          - podrobnější informace o souborech a instalaci
|---- docs
|  |---- other
|  +---- thesis          - práce a její zdrojové soubory
|    +---- src
|---- photos
|  |---- house           - fotografie domu a žaluzií
|  +---- making-of       - fotodokumentace výroby meteostanice a ovladače žaluzií
+---- src
  |---- mcu              - zdrojové kódy pro mikrokontroléry
  |  |---- ArduinoIDE-config - podpora pro ESP-WROOM-02
  |  |---- dev-code        - testovací kód pro vývoj
  |    |---- esp8266-BMP280
  |    |---- esp8266-TLS2561
  |    |---- esp8266-wind_direction
  |    +---- esp8266-windspeed
  |  |---- libraries      - zdrojové kódy knihoven
  |    |---- AS3935
  |    |---- SEN-08942     - knihovna pro senzory meteostanice
  |    +---- Somfy         - knihovna pro ovládání žaluzií
  |  +---- main-code       - software pro mikrokontroléry
  |    |---- esp8266-blind_controller
  |    +---- esp8266-station
  |---- automatization    - zdrojové kódy automatizačního programu
  |  +---- concept         - návrh automatizačního programu napsaný v~javě
+---- other
  |---- label             - etiketa odevzdaného disku
  +---- raspberry-pi-blind-controller - ovladač žaluzií pro Raspberry Pi
```